

УДК 004.413

Информационно-программный комплекс космической системы наблюдения за морскими объектами

Н.В. Малова

Аннотация

Показана актуальность решения задачи наблюдения за морскими объектами и создания единого информационного центра. Приведено описание метода создания информационно-программного комплекса (ИПК) космической системы. Для хранения информации предложена и описана структура объектной базы данных (БД), представлены принципы решения задачи наблюдения средствами космических аппаратов. Описаны результаты разработки с применением предложенной технологии.

Ключевые слова

информационно-программный комплекс; информационная технология проектирования; база данных; система управления базами данных; информационно-управляющая система; фреймворк.

1. Постановка задачи

На сегодняшний день все большую актуальность приобретает мониторинг перемещения судов в акватории мирового океана. Существуют некоторое количество центров наблюдения, контролирующих различные объекты, однако информация между ними не синхронизируется, нет общей картины обстановки. Возникает необходимость создания общего Центра сбора и обработки информации (ЦСОИ), куда бы стекались все данные по морским объектам, которые будут храниться в определенном виде с возможностью быстрого доступа.

Источники информации по обстановке мирового океана могут быть различные, кроме средств связи судов и других способов предлагается использовать космическую составляющую. Привлечение космических средств дает следующие возможности:

- независимая информация о положении морских объектов;
- контроль обстановки прибрежных зон;
- поиск пропавших судов (не выходящих на связь);

- возможность мониторинга опасных зон или заданного района (например, наблюдение за районом перемещения пиратских судов).

Основные районы с особым режимом плавания отражены в лоциях и Сводных описаниях районов океанских зон. В этих районах опасности, подстерегающие суда - статические. При возникновении динамических опасностей таких как: цунами, начавшиеся военные действия и т.д. необходимо решать задачу обеспечения безопасности плавания путем своевременного информирования и оценки времени возвращения судов в ближайшие безопасные зоны.

В настоящее время контролю подлежат свыше десятка тысяч морских объектов различного назначения, находящихся в акватории мирового океана. Так как для принятия решения и прогнозирования ситуации нужны не только данные в текущий момент времени, но и за предыдущее время, то необходимое количество точек мирового океана, подлежащих контролю, может быть около миллиона.

Для создания информационно-программных систем, оперирующих значительными объемами данных, целесообразно использование информационной технологии проектирования (ИТП) [1,2,3]. ИТП определим как процесс разработки, выполняемый при следующих условиях:

- применение баз данных (БД), как именованной совокупности данных, отражающей состояние объектов и их отношений в рассматриваемой предметной области, а также систем управления базами данных (СУБД), как совокупности языковых и программных средств для создания и ведения БД;
- способ проектирования состоит в создании информационно-логической модели данных, дополняемой функциями преобразования данных (функциональным программным обеспечением (ФПО));
- применение средств обеспечения функционирования и разработки.

Согласно принятой концепции, основой этой технологии являются данные, которые должны быть организованы в виде БД с целью адекватного отображения изменяющегося реального мира и удовлетворения потребностей пользователей.

Анализ разработанных программных комплексов обработки информации и управления показывает, что значительная программного кода выполняют поиск, выборку и упорядочивание данных. Проведя анализ информации на этапе проектирования и, создав упорядоченную структуру в виде БД, можно воспользоваться предоставляемыми средствами СУБД,

которые обеспечивают большинство функций работы с данными. Недостающие функции разрабатываются в виде приложений и библиотек.

2. Описание информационно-управляющей системы наблюдения за морскими объектами

Рассмотрим информационно-управляющую систему (ИУС), которая решает задачу наблюдения за морскими объектами: кораблями различного назначения, районами путем использования информации с космических аппаратов (КА), являющихся средствами наблюдения (СН) (Рис. 1).

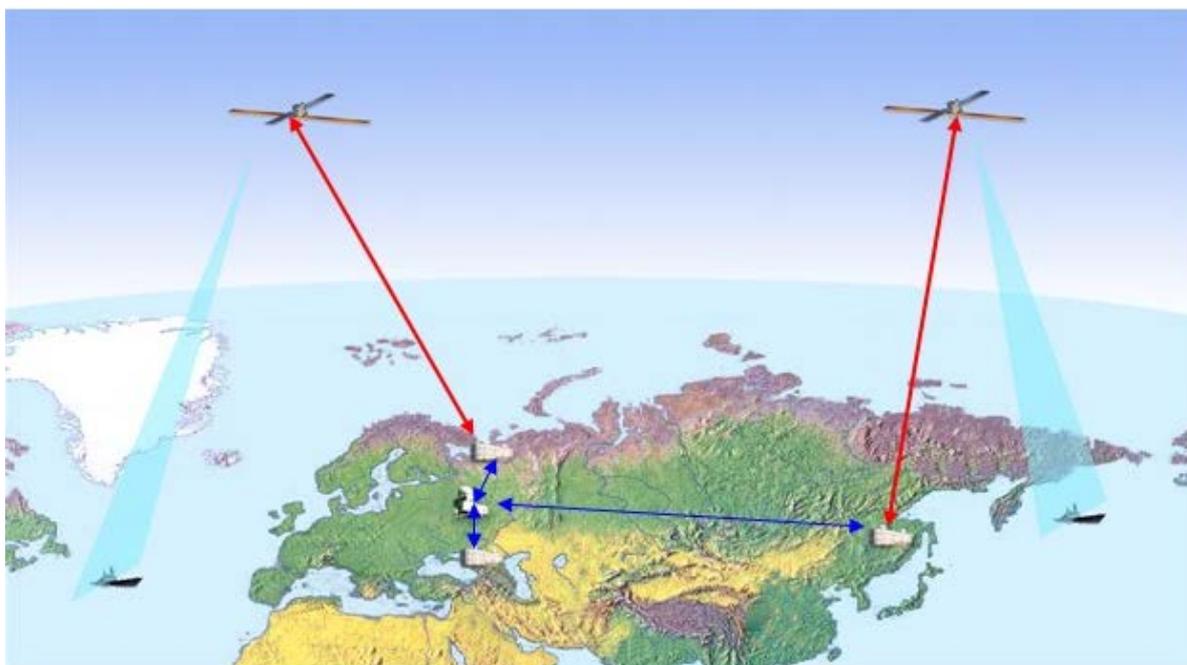


Рис. 1 Схема сбора информации с КА о местоположении кораблей и доставки её потребителям

На схеме изображены две орбитальные группировки со своими пунктами управления и пунктами сброса информации. Бортовая аппаратура обнаружения (БАО) КА, включаемая в соответствии с данными прогноза движения КА, определяет параметры объектов. Информация передается на пункты сброса и доставляется в центр обработки, в котором рассчитываются очередные зоны осмотра и данные по обстановке, и передает их активным потребителям.

Для приведенной системы характерны следующие особенности:

- работа в реальном масштабе времени;
- специфические требования по надежности и безопасности функционирования;
- непрерывный режим функционирования;

- возможное отсутствие оператора на этапах приема и обработки информации;
- операторный режим формирования и выдачи данных обстановки активным потребителям.

Реализация подобной ИУС (Рис. 2) требует создания информационно-программного комплекса (ИПК), содержащего:

- базы данных с информацией по кораблям, космическим аппаратам, районам и т.п.;
- процесс обработки данных, состоящий из:
 - процесса обработки данных о текущем положении кораблей и планирования указаний космическим системам;
 - процесса приема и обработки информации об объектах от различных источников;
- набор форм оператора, представляющих собой интерфейс для настройки и мониторинга ИУС, отправки сообщений.



Рис. 2 Общая схема обмена информацией

Персонал, обслуживающий ИУС, обеспечивает автоматизированное управление системой на основании информации, предоставляемой ему в составе интерфейса пользователя на рабочих местах (РМ) операторов. Для рассматриваемой системы необходимы следующие функции:

- формирование задания на работу в виде назначений объектов исполнения и наблюдения средств наблюдения на основании приходящих заданий;
- планирование: расчет и оптимизация плана работы взаимодействующих систем;
- корректировка и утверждение сформированной выходной информации;

- обеспечение безопасности информации: формирование и контроль доступа персонала к информации;
- администрирование базы данных: обеспечение актуальности, целостности и сохранности информации в БД;
- прием информации, контроль поступающих от внешних абонентов данных на соответствие полноты и достоверности;
- выдача информации внешним абонентам по электронной почте;
- ведение обстановки с применением электронной карты;
- обеспечение гидро-метео информацией.

Этот далеко не полный перечень функций показывает необходимость создания распределённого ИПК, в котором функции разделены между операторами и абонентами.

3. Метод создания ИПК

3.1. Создание системы обработки данных

Из вариантов создания среды, поддерживающей подобную структуру, наиболее предпочтительным является метод, применяемый в веб-технологиях и, прежде всего, по причине доступности решений, наличия готовых программ для комплектации изделия и выполнению требований по надежности и качеству функционирования [4].

HTTP (сокр. от англ. *HyperText Transfer Protocol* — «протокол передачи гипертекста») — протокол прикладного уровня передачи данных (изначально — в виде гипертекстовых документов). Основой HTTP является технология «клиент-сервер», то есть предполагается существование потребителей (клиентов), которые инициируют соединение и посылают запрос, и поставщиков (серверов), которые ожидают соединения для получения запроса, производят необходимые действия и возвращают обратно сообщение с результатом.

HTML (от англ. *HyperText Markup Language* — «язык разметки гипертекста») — стандартный язык разметки веб-документов. Большинство веб-страниц создаются при помощи языка HTML. Язык HTML интерпретируется браузерами и отображается в виде документа, в удобной для человека форме.

Предложенный метод состоит в использовании связки программного обеспечения (Рис. 3):

- HTTP-сервер, установленный на выделенной машине в сети (Сервер) (принимает от клиентов HTTP запросы и выдает HTTP ответы);

- система управления базами данных (СУБД), где сервер СУБД может совпадать с машиной Сервер;
- HTML-браузер, установленный на машине клиента и таких машин может быть произвольное количество.

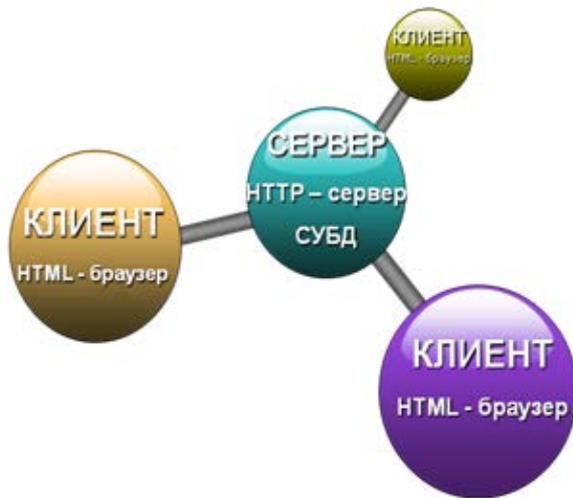


Рис. 3 Метод создания ИПК

Особенностями данного метода являются:

- обмен данными осуществляется по протоколу HTTP;
- СЕРВЕР представляет собой связку: стандартный веб-сервер + БД + СУБД + ФПО, там происходит процесс авторизации, выполнение запросов, обработка ошибок;
- КЛИЕНТЫ - стандартные веб-браузеры, на клиентах осуществляют только отправка запросов и воспроизведение ответов, присланных сервером, в виде HTML-страниц.

Применение метода связано с распределенным характером системы и необходимостью интеграции возможностей и ресурсов всех участников системы, которые нуждаются в согласованной информации, а также необходимостью сокращения трудозатрат на разработку при сохранении функций и качеств систем.

Предлагается реализация метода с использованием следующих средств ИТП **КорЕх**:

- веб-сервер – apache;
- фреймворк PHP **Kohana** (в качестве слоя обработки данных, скриптовый серверный язык для создания динамических веб-приложений);
- интерпретатор серверных скриптов PHP;
- СУБД **Postgresql** (Линтер-ВС);
- фреймворк JavaScript **ExtJs** (для разработки пользовательских интерфейсов, поддерживает технологию AJAX, анимацию, работу с DOM, реализацию таблиц, вкладок, обработку событий);
- интерпретатор клиентских скриптов JS;
- базовый веб-клиент Mozilla FireFox;
- платформа Linux (MCBC) или Windows NT.

3.1.1. PHP фреймворк Kohana

Фреймворк (англ. *framework*) — в информационных системах структура программной системы; программное обеспечение, облегчающее разработку и объединение разных компонентов большого программного проекта. В отличие от библиотек, которые объединяют набор подпрограмм близкой функциональности, фреймворк содержит в себе большое количество разных по назначению библиотек.

Особенностью фреймворков для языка PHP является возможность сделать путь обращения к файлам, функциям и параметрам одновременно неявным и удобным для пользователя.

Основное преимущество использования фреймворков – это использование готового каркаса приложения, построенного по принципу Model-View-Controller (MVC). Принцип MVC (Модель–Представление–Поведение)—это способ объединить три базовые функции современного информационного портала:

- функции работы с БД;
- отображение информации;
- программная логика и обработка запросов.

При этом происходит разделение данных, их представления и обработки на отдельные компоненты, что позволяет строить приложения по принципу конструктора, где существенное значение для взаимодействия играет программный интерфейс компонентов.

Эти три базовых функции как раз представлены тремя компонентами модели MVC, моделью, представлением и контроллером.

Model (модель) – это работа с базой данных, файловыми хранилищами и другими источниками данных.

Модель предоставляет данные для представления и реагирует на запросы Контроллера, изменяя своё состояние. Фактически моделью является обычный класс для работы с разными данными.

View (представление) отвечает за вывод информации (пользовательский интерфейс). По сути, вид, в котором выводятся данные обработанные моделями.

Controller (поведение) – это логика системы. Интерпретирует данные, введенные пользователем, и информирует модель и представление о необходимости соответствующей реакции.

Работу MVC можно представить следующим алгоритмом (Рис. 4):

1. Пользователь взаимодействует каким-либо образом с пользовательским интерфейсом (нажатие мыши).
2. Контроллер получает входящее событие и обрабатывает его в вид, понятный для модели.
3. Контроллер уведомляет модель о действии пользователя, после чего возможно изменение состояние модели
4. Возможно самостоятельное обновление представления. Но чаще всего контроллер оповещает представление о том, что модель изменилась и необходимо обновление представления, или об этом сообщает сама модель.
5. Представление получает данные для обновления, и пользовательский интерфейс ждет дальнейших действий пользователя.

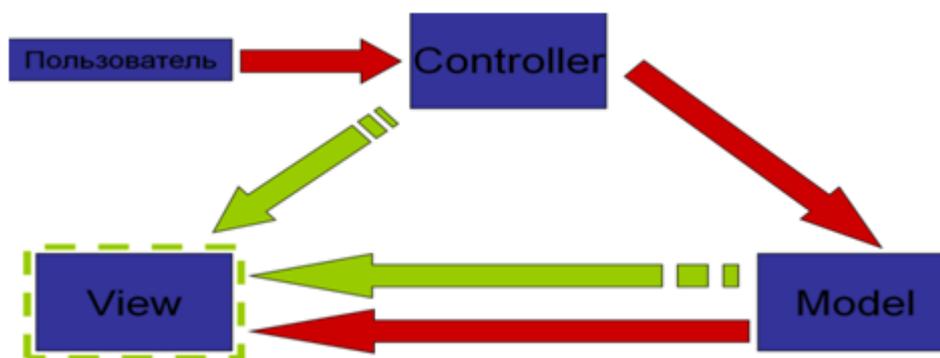


Рис. 4 Общее представление о работе MVC

3.1.2. JavaScript фреймворк ExtJS

Библиотека ExtJS написана на JavaScript и работает во всех популярных сейчас браузерах, предназначена для создания сложных и насыщенных интерфейсов. Она предоставляет разработчику целый набор графических компонентов, от тривиальных кнопок и расширенных элементов обычных HTML-форм, до сложнейших компонентов вроде таблиц и деревьев. В библиотеке также есть достаточно много невидимых пользователю компонентов, которые и обеспечивают работу того, что мы видим на экране. Это и получение данных с сервера в фоновом режиме (в формате JSON или XML), обновление частей страницы, локальные хранилища данных, и многое другое.

Каждый компонент в ExtJS позволяет себя конфигурировать путем настройки разнообразных опций, генерирует множество событий в ответ почти на любое изменение своего состояния, а также гибко настраивается через задание необходимых свойств оформления.

Библиотека ExtJS является стабильным и качественным продуктом промышленного уровня. Развивается и поддерживается специально созданной компанией. Наличие мощных

компонентов для работы с данными позволяет строить разные аналитические и управленческие системы и сервисы для управления проектами. Идеально подходит для решения создания графического интерфейса в соответствии с поставленной задачей.

3.1.3. СУБД PostgreSQL

Для решаемых задач было выбрано сертифицированное средство СУБД Линтер-ВС и его аналог PostgreSQL. Такой выбор обеспечивает создание информационных компонент с выполнением требований по безопасности информации [5].

База данных под управлением СУБД Линтер предназначена для хранения статической и динамической информации, используемой в процессе моделирования, обмена данными. База данных имеет фиксированную структуру.

3.2. Представление и хранение данных

Согласно ИТП информация представлена в виде БД объектной структуры, полностью отражающей все необходимые свойства и функции объектов [6].

Описание объекта представляет собой набор таблиц (Рис. 5):

- **головная информация** (Данные), специфичная для экземпляра объекта;
- **типовая информация** (\$ТИП), характеризующая серию объектов данного типа;
- **значения** (\$ЗНАЧ) - чаще всего цифровые, реже – символьные параметры, которые могут быть при описании объекта, относящиеся как к отдельному экземпляру, так и ко всем экземплярам типа;
- **состояние** (\$СОСТ) - описание состояния объекта на определённое время;
- **словарь** (\$СЛ) - содержит словарные термины, используемые во всех компонентах описания объекта;
- **соединение** (\$СДК), связывающая данный объект с другими объектами;
- **фильтр** (\$ФЛ) – для создания списков и фильтров возможных значений.

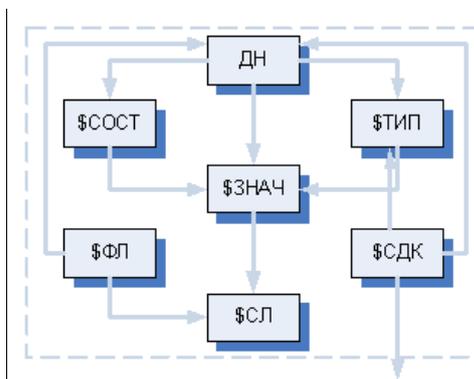


Рис. 5 Структура типового объекта БД

Все таблицы БД имеют свой номер, который хранится в соответствующей таблице ТАБЛ. Рассмотрим представление объекта «Космический аппарат» (КА) (Рис. 6).

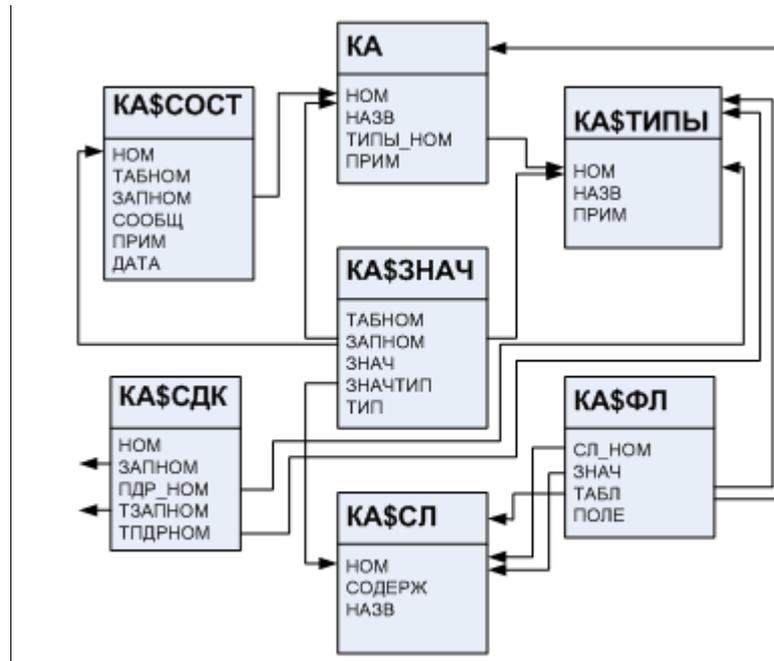


Рис. 6 Представление объекта КА

В головной таблице КА содержатся номера, названия КА (например, Ресурс ДК) и ссылка на их тип в таблице типов КА_ТИПЫ (Ресурс ДК), в которой также содержится название типа, а также примечание.

В таблице значений содержатся как индивидуальные параметры КА (нап-р угол обзора), так и параметры, общие для КА одного типа (например, пункт сброса или тип бортовой аппаратуры). Значение может быть соответственно привязано по полю ТАБНОМ к таблице КА или КА_ТИПЫ по определенному номеру записи ЗАПНОМ. В таблице также указывается значение (ЗНАЧ), его размерность (ЗНАЧТИП) (ссылка на словарь КА_СЛ) и тип (ТИП).

Оперативные (динамические) параметры привязаны к состоянию КА на определенное время в таблице КА_СОСТ ,это, например, параметры орбиты :эксцентриситет, угол наклона, долгота восходящего узла и др.) В таблице отображаются: дата состояния (параметров) (ДАТА), с помощью полей ТАБНОМ, ЗАПНОМ, НОМ производится привязка к таблицам КА, КА_ЗНАЧ, также отображается само сообщение (СООБЩ) и примечание (ПРИМ).

Таблица КА_СЛ содержит номер записи (НОМ) для связи с другими таблицами, различные названия, параметры и термины (НАЗВ, СОДЕРЖ), на которые ссылаются таблицы значений и фильтров.

В таблице КА_ФЛ отражаются фильтры и списки n-р список параметров с единицей измерения градусы и т. п. СЛ_НОМ – номер в таблице КА_СЛ; ТАБЛ, ПОЛЕ – номер таблицы и записи; ЗНАЧ – значение фильтра.

Объект ДК_ГР позволяет описать «статические данные» гражданского корабля с помощью 6-и связанных таблиц:

- ДК_ГР - головная, содержит название корабля и ссылки на тип корабля и его состояние готовности

Прочие таблицы являются служебными по отношению к головной и имеют составное имя:

- ДК_ГР\$ТИП - содержит данные, характеризующие тип постройки корабля: шифр, страна постройки, категория корабля (надводный, подводная лодка, вспомогательное судно);
- ДК_ГР\$СЛ - словарь, содержит словарные термины, их содержание, обозначение и раздел словаря;
- ДК_ГР\$СОСТ - содержит словарные термины по состоянию готовности корабля на определённое время;
- ДК_ГР\$СДК - позволяет связать объект ДК_ГР с другими объектами; содержит номер объекта ДК_ГР, имя связуемой таблицы и номер записи в связуемой таблице;
- ДК_ГР\$ЗНАЧ-содержит значения параметров, используемых в таблицах объекта ДК_ГР; указывается имя таблицы, номер записи таблицы, значение параметра и его словарные атрибуты: название, размерность и тип данных.

Объект ТСГ_ГР позволяет описать «динамические данные» объектов. В частном случае для кораблей такими данными могут быть параметры движения и состав группы кораблей. Предполагается, что группа кораблей имеет уникальное название. Параметры движения относятся ко всей группе. Одиночные корабли представляются как частный случай группы путём указания типа группы. Такая инфологическая модель позволяет представить, в том числе и конфигурацию группы на географической карте с помощью записи значений, определяющих местоположение корабля относительно флагмана. Объект ТСГ_ГР состоит из 6-и связанных таблиц.

- ТСГ_ГР - головная, содержит данные на каждое сообщение о группе: дату/время, название источника сообщения, количество объектов в группе и ссылки на тип группы и на словарь, где хранятся словарные термины по типу

источника сообщений, виду деятельности группы и положению (надводное, подводное).

Служебные таблицы:

- ТСГ_ГР\$ТИПЫ - содержит название и ссылку на таблицу ХРАН, где хранится символическое изображение и шифр типа группы;
- ТСГ_ГР\$СЛ - словарь, содержит словарные термины, их содержание, обозначение и раздел словаря;
- ТСГ_ГР\$СОСТ - содержит данные на сообщения о готовности группы и нестандартных ситуациях с возможностью указания конкретного объекта группы;
- ТСГ_ГР\$СДК - позволяет связать объект ТСГ_ГР с другими объектами; содержит номер объекта ТСГ_ГР, имя связуемой таблицы и номер записи в связуемой таблице;
- ТСГ_ГР\$ЗНАЧ - содержит значения параметров, используемых в таблицах объекта ТСГ_ВМФ; указывается имя таблицы, номер записи таблицы, значение параметра и его словарные атрибуты.

Объект СТАТ позволяет описать статические объекты: пункты базирования, береговые средства наблюдения, полигоны и др. Эти объекты имеют название, относятся к определённому типу, определяются по географическим координатам, находятся в каком-либо состоянии и включают в себя другие объекты, такие как радиолокационные и радиотехнические средства, личный состав и т.д.. Объект СТАТ состоит из 6-и связанных таблиц.

- СТАТ-головная, содержит название и номер объекта, ссылки на таблицу СТАТ\$ТИПЫ и на таблицу ХРАН, где хранится символическое изображение и шифр объекта;

Служебные таблицы:

- СТАТ\$ТИПЫ - содержит название типа объекта и ссылку на таблицу ХРАН, где хранится символическое изображение и шифр типа объекта;
- СТАТ\$СЛ - словарь, содержит словарные термины, их содержание, обозначение и раздел словаря;
- СТАТ\$СОСТ - содержит информацию о вводе в эксплуатацию объекта, изменении режимов использования и других аспектах жизнедеятельности объекта с возможностью указания конкретных цифровых данных;

- СТАТ\$СДК - позволяет связать объект СТАТ с другими объектами; содержит номер объекта СТАТ, имя связуемой таблицы и номер записи в связуемой таблице;
- СТАТ\$ЗНАЧ - содержит значения параметров, используемых в таблицах объекта СТАТ; указывается имя таблицы, номер записи таблицы, значение

Объект РАЙОН позволяет описать географические районы произвольной конфигурации. Местоположение района указывается широтами и долготами точек периметра района. Количество точек не ограничено. Объект РАЙОН состоит из 6-и связанных таблиц.

- РАЙОН - головная, содержит название района, шифр и ссылки на тип района и на таблицу ХРАН, где хранится символическое изображение района.

Служебные таблицы:

- РАЙОН\$ТИПЫ - содержит название типа и ссылку на таблицу ХРАН, где хранится символическое изображение и шифр типа РАЙОНА;
- РАЙОН\$СЛ - словарь, содержит словарные термины, их содержание, обозначение и раздел словаря;
- РАЙОН\$СОСТ - содержит данные на район: дату/время, словарные термины по готовности района(закрит, закрыт для рыболовных судов и т. д.) ;
- РАЙОН\$СДК - позволяет связать объект РАЙОН с другими объектами; содержит номер объекта РАЙОН, имя связуемой таблицы и номер связуемого объекта в связуемой таблице;
- РАЙОН\$ЗНАЧ - содержит значения параметров, используемых в таблицах объекта РАЙОН; указывается имя таблицы, номер записи таблицы, значение параметра и его словарные атрибуты: название, размерность и тип данных.

Согласно структуре типового объекта представлены все объекты БД в т. ч.: ВМФ, ТСГ_ВМФ, ВМС, ТСГ_ВМС и др.

Разработанная структура предназначена, прежде всего, для БД, описывающих функционирование систем, содержащих динамические и статические объекты, объединённые в структуры. Технология обеспечивается разработанными средствами БД и обладает следующими достоинствами:

- легкость модернизации при проектировании; БД собирается из типовых объектов; состав выдаваемой и хранимой информации меняется с помощью предусмотренных таблиц специального вида;

- динамичность при сопровождении; БД позволяет учесть практически все новые требования по структуре и составу информации, выдвигаемые пользователями;
- эффективность хранения информации; в БД нет запасенных мест для будущей информации, места появляются при её поступлении.

Использование БД в ИПК дает возможность хранения различного рода данных в упорядоченной структуре, что обеспечивает оперативное получение информации для расчетов и различной обработки, а также вывод информации различной компоновки, представляющей собой всевозможные таблицы и отчеты.

4. Принципы решения задачи наблюдения средствами КА

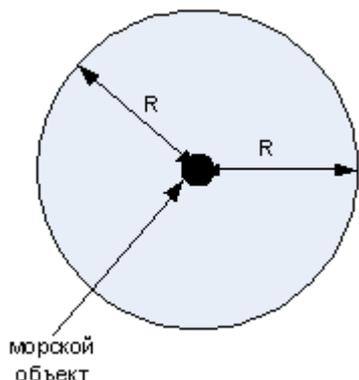
Для решения задачи наблюдения реализованы следующие функции обработки данных:

- расчет трасс движения КА на заданном интервале с использованием аналитического прогнозирования;
- определение географических координат пересечения трасс КА с зонами возможного нахождения объектов;
- выборка информации из БД по текущим параметрам КА, техническим характеристикам аппаратуры наблюдения для расчета плана работы КА;
- прогноз движения объекта с задаваемыми вероятностными параметрами для получения трассы движения объекта;
- режим ускоренного времени с отображением перемещений объекта на карте;
- формирование заявок на осмотр для взаимодействующих систем в формате XML;
- взаимодействие с источниками и перегрузка информации средствами SQL [5] с созданием локальных БД для обмена данными и др.

Средствами СУБД осуществляется выборка информации из БД, являющейся входными данными для функций, производится обработка этих данных и вставка в БД полученных результатов в соответствии со структурой.

Таким же образом реализуется аналитическая модель орбитального движения космических аппаратов, оснащенных БАО различных типов (оптическая, радиотехническая, радиолокационная), путем расчета точек положения КА по выбранным характеристикам на определенный момент времени. По объектам наблюдения исходными данными является координатно-объектовая информация, по которой определяется область возможного положения

(ОВП) на время моделирования (Рис. 6), она представляет собой круг с радиусом, равным разности времени моделирования и времени последней координатной информации в БД по данному объекту, умноженной на скорость морского объекта. Если информации по скорости нет, то берется среднее значение, равное 20 узлам.



$$R = (t_{ТЕК} - t_{БД}) \cdot V \quad (1),$$

где:

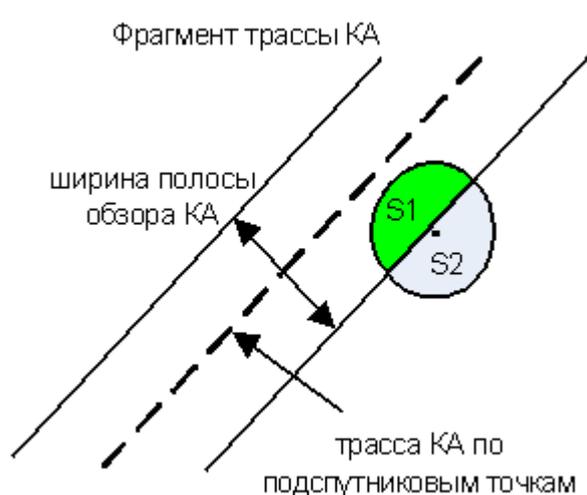
R – радиус ОВП морского объекта;

$t_{ТЕК}$ – время моделирования;

$t_{БД}$ – время последней координатной информации в БД по объекту наблюдения;

Рис. 6 ОВП морского объекта V – скорость объекта наблюдения.

Далее производится расчет пересечения трассы КА, а именно области, соответствующей ширине полосы обзора и типу БАО, с рассчитанными ОВП (Рис. 7). Рассчитывается коэффициент накрытия путем отношения площадей полосы накрытия к области ОВП и, чем меньше ОВП (чем чаще поступают сообщения о положении ОН), тем ближе к 1 будет коэффициент накрытия и тем выше будет вероятность реального обнаружения космическим аппаратом [7].



$$K_{НАКР} = S_1 / S_2 \quad (2),$$

где:

$K_{НАКР}$ – коэффициент накрытия;

S_1 – площадь пересечения полосы обзора и ОВП ОН;

S_2 – площадь ОВП ОН

Рис. 7 Определение коэффициента накрытия

С целью более точного вычисления ОВП морских ОН и составления прогнозов разработана и реализована модель их движения (Рис. 8), где принцип следующий:

- задается начальное положение ОН (последняя информация в БД по ОН);

- выполняется прогноз движения разыгрыванием курса и скорости с заданными вероятностными параметрами;
- определяются координаты ОН (широта и долгота) в точке обнаружения средствами БАО.

Все смоделированные значения скорости и курса корректируются в соответствии с возможными допустимыми значениями этих параметров. Например, для скорости установлены границы от 0 до 30 узлов и, в случае выхода за верхний или нижний предел, присваивается соответствующее граничное значение. Таким образом, моделирование движения ОН вырезает из круга ОВП «сектор», соответствующий курсу морского объекта с учетом среднеквадратического отклонения, а радиус составит произведение скорректированной случайной скорости с учетом среднеквадратического отклонения на время.

Полученная точка положения морского объекта проверяется на принадлежность полосе обзора КА в данный момент времени и в случае положительного результата формируется заявка на осмотр.

По смоделированным трассам КА и движению морских объектов, синхронизированным во времени (Рис. 9), определяются временные параметры начала и конца осмотра ОН и передачи данных осмотра на наземные пункты приема информации, координаты и радиусы сброса которых для различных космических систем хранятся в БД.

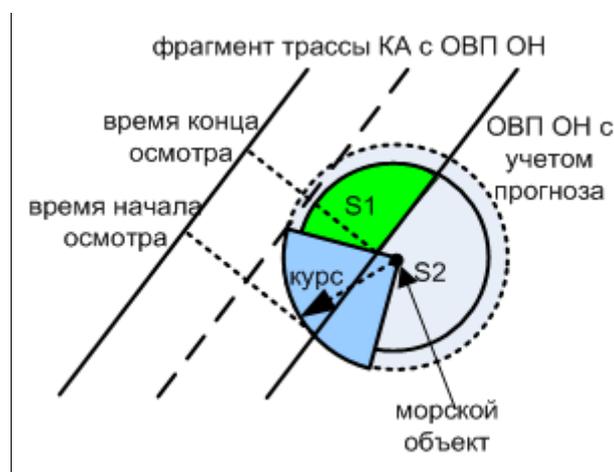


Рис. 9 Уточненная ОВП ОН и определение интервалов наблюдения

Положительные результаты осмотра по сформированным заявкам поступают в БД, образуя трассы движения ОН (Рис. 10) и области их возможных положений, по которым рассчитываются новые заявки на следующий цикл работы системы.



Рис. 10 Полученные трассы ОН

Реализация описанных выше функций позволяет получить интервалы контроля объектов для различных исходных данных: количество КА, параметры орбит, характеристики аппаратуры наблюдения, трассы движения объектов.

5. Результаты разработки

Результатом разработки является создание ИПК космической системы наблюдения за морскими объектами на базе предложенной ИТП КоРЕх, где реализовано следующее:

- распределенная ИС с архитектурой «Клиент-Сервер» (использование веб-сервера **apache**);
- многопользовательская система с разграничением прав доступа к тем или иным функциям; согласно выполняемым функциям каждому пользователю присваивается роль или несколько ролей, а роль представляет собой список доступных приложений, пользовательских форм и т.д. (использование фреймворка Kohana, СУБД Postgres);
- информация хранится в БД объектной структуры с возможностью отображения в различном составе, поиска, редактирования, добавления/удаления, выборки; хранению и отображению подлежит не только текстовая информация – реализован каталог фотоизображений (использование СУБД Postgres, средств SQL, фреймворка ExtJS для создания пользовательского интерфейса);

- отображение различной информации в виде как двумерных, так и трехмерных графиков, что представляет значительное удобство при анализе больших объемов числовых данных, поиске ошибок (использование PHP библиотеки JGraph);

- функции и интерфейсные формы пользователя для решения задачи планирования осмотров КА с различными видами БАО (радиотехнической, радиолокационной, оптической):

- обработка заданий на осмотры и назначение всех участвующих объектов;

- расчет матрицы возможностей осмотров, которая представляет собой таблицу с названиями объектов наблюдения, средств наблюдения, временами начала и конца обзора и сброса, коэффициентами покрытия, метеоусловиями в момент осмотра и т.п.;

- отбор рассчитанных возможностей осмотра по различным критериям (коэффициентам покрытия, времени сброса, метеоусловиям по типу БАО);

- обмен данными с источниками информации.

6. Заключение

Предложенная в конкурсной работе ИТП КоРЕх создавалась и совершенствовалась по мере разработки в подразделении сложных программных комплексов. Технология реализована при создании ИПК космической системы наблюдения за морскими объектами. Анализируя результаты применения ИТП КоРЕх, можно отметить следующие преимущества:

- увеличение скорости разработки и отладки комплекса;
- уменьшение трудозатрат на проектирование, разработку, создание интерфейса пользователя за счет использования структурированных данных и интеграции готовых решений в средствах автоматизации проектирования;
- кроссплатформенность – возможность реализации сервера и браузера на разных платформах;
- простота обеспечения информационного и управляющего взаимодействия между приложениями комплекса;
- соответствие мировым стандартам форматов обмена данными, что обеспечивает подключение новых источников информации и потребителей;
- получение заданных временных параметров работы подсистем и комплекса в целом.

Особенностью предложенной ИТП является универсальность использования при создании информационных центров распределенного характера (к примеру, проведен анализ,

давший положительный результат, о возможности применения ИТП KoPEx при создании ИПО для Информационно-аналитического центра астероидно-кометной безопасности).

Библиографический список

1. *Бозм Б.* Инженерное проектирование программного обеспечения. - М.: Радио и связь, 1985.
2. *Boehm B.W.* Software cost estimation with COCOMO II. Prentice Hall PTR. - New Jersey, 2000.
3. *Луцаев В.В.* Программная инженерия. Методологические основы. – М.: ТЕИС, 2006 – 608 с.
4. *Можар Т.Е.* Технология разработки систем имитационного моделирования//
Высокие интеллектуальные технологии и инновации в образовании и науке. Материалы XVII Международной научно-методической конференции 11-12 февраля 2010 года. Том 3. Национальная безопасность. - Спб.2010. С. 123-131.
5. *Грофф Дж., Вайнберг П.* SQL: Полное руководство/Пер. с англ. – 2-е изд., перераб. и доп.– Киев: Издательская группа BHV, 2001. – 816 с.
6. *Гребёнкин В.Ф., Мисник В.П.* Методы формализации задач с применением объектной базы данных мобильной структуры // Космические информационно-управляющие системы. - М.: ФГУП «ЦНИИ «Комета», 2006. Вып. 1.
7. *Никулин Е.А.* Компьютерная геометрия и алгоритмы машинной графики. – СПб.: БХВ-Петербург, 2003 – 560 с.

Сведения об авторах

МАЛОВА Наталья Викторовна, инженер-программист 1й категории

ФГУП «ЦНИИ «Комета», аспирант 4 года обучения; 115280, Москва, ул. Велозаводская, д. 5;
р. тел.: +7 (495) 674 09 83, с. тел.: +7 (903) 512 72 39;

e-mail: info@cniikometa.ru, nata.pochta@bk.ru