

УДК 621.165

А.В. БОЙКО, д-р техн. наук, проф., Ю.Н. ГОВОРУЩЕНКО, канд. техн. наук,  
А.П. УСАТЫЙ, канд. техн. наук

*Национальный технический университет «Харьковский политехнический институт»*

### **ОСОБЕННОСТИ ИНФОРМАЦИОННОГО ОБМЕНА В РАМКАХ ЕДИНОГО ИНФОРМАЦИОННОГО ПРОСТРАНСТВА САПР «ТУРБОАГРЕГАТ»**

Розглянено механізми обміну інформацією між елементами єдиного інформаційного середовища САПР «Турбоагрегат», показана висока ефективність, універсальність та надійність приведених підходів та алгоритмів передачі даних між різними проектними процедурами і функціями САПР, а також експорту-імпорту даних з/в зовнішніх джерел. Показано вплив механізмів обміну даними на формування і організацію обчислювальних процесів в задачах аналізу та оптимального проектування проточних частин осевих турбін.

The mechanisms of information exchange between elements of the “Turboaggregate” CAD system unified information space are examined. High efficiency, versatility and reliability of adduced approaches and algorithms of data exchange between different project procedures and CAD functions as well as data import/export from/to external sources are shown. The influence of data exchange mechanisms on formation and organizing of calculation processes related to the problems of axial turbine’s flow-path analysis and optimal design is considered.

Одним из современных направлений развития САПР «Турбоагрегат» является направление, базирующееся на реализации идеи единого информационного пространства (ЕИП) [1], как наиболее перспективное с точки зрения требований современных информационных технологий. Понятно, что ЕИП САПР «Турбоагрегат» предназначено для размещения в нем информации о реальных объектах проектирования (турбина, цилиндр, ступень, ...) в виде соответствующих им неких информационных аналогов (ИА), а также основных технологических (сервисно-административные функции) и прикладных (функции алгоритмов и проектные процедуры) компонент, предназначенных для обработки и управления этими ИА. Все вместе они являются элементами ЕИП.

Прежде чем приступить к рассмотрению механизмов и алгоритмов управления обменом информацией между внутренними параметрами основных компонент ЕИП и элементами ЕИП соответствующими предметным и размерным сущностям реальных объектов проектирования, рассмотрим предварительно структуру атрибутов элементов ЕИП. Понимая, что ЕИП состоит из элементов, которые, в свою очередь, объединены в соответствующие иерархические структуры (Объекты проектирования, Справочники, Атласы, ...), а также включены в различные модели (Управления состоянием, Допуска пользователей, Сценариев процессов, ...) и могут принадлежать к различным информационным слоям (Предварительные исследования, Техническое задание, ... Рабочий проект, Технология, Производство, ... Эксплуатация, Демонтаж, Утилизация и т.д.), возникает одно важное требование ко всем элементам ЕИП – это требование однозначного позиционирования любого элемента в самом ЕИП и в структурах, моделях, слоях и т.п., которые, также являются элементами ЕИП. Анализируя возможные методы и формы организации такого позиционирования, можно сделать вывод о том, что наиболее эффективной является методология, основанная на индексировании элементов, нашедшая широкое развитие и применение в системах управления базами данных (СУБД) [3], где в качестве элемента выступает запись в

таблице, а в качестве индекса – уникальный ключ, генерируемый средствами СУБД и используемый для идентификации этой записи в таблице. Учитывая, что СУБД является составной частью информационного обеспечения САПР «Турбоагрегат» [1], воспользуемся ее возможностями, организовав средствами СУБД единый реестр элементов (ЕРЭ) ЕИП САПР. Где каждый элемент представляет собой запись в соответствующей таблице, содержащей кроме уникального ключа-индекса информацию об идентификаторе и классе элемента, принадлежности его к структуре, модели, слою, состоянию, правах доступа и др., обеспечивая при этом уникальность индекса элемента и единое пространство индексов всех элементов в системе. Вся эта информация, характеризующая элемент в ЕИП САПР, содержится в атрибутах элемента, состоящих из нескольких блоков, сформированных с учетом объектно-ориентированного подхода [2]:

- **Index** {ID;idC;idS;idM;idL}
- **View** {iV(Font;Size;B;I;cT;cB);flcon;Mask(Show;Down;Read;Write;Copy;Clone;Del;...)}
- **Lang** {Ru;Ukr;An;Pol;Ital;....}
- **Value** {Type;Long;Dec;Min;Init;Max;idUnit;kSys}
- **Struct** {Parent(ID<sub>1</sub>;...ID<sub>n</sub>); Sub(ID<sub>1</sub>;...ID<sub>n</sub>)}
- **mModel**{Parent(ID<sub>1</sub>;...ID<sub>n</sub>); Sub([ID;iV;Mask]<sub>1</sub>;... [ID;iV;Mask]<sub>n</sub>); Controller(ID<sub>1</sub>;...ID<sub>n</sub>)}
- **iModel** {Parent(ID<sub>1</sub>;...ID<sub>n</sub>); Sub([ID;idL]<sub>1</sub>;... [ID;idL]<sub>n</sub>)}
- **pModel** {In([ID;Link]<sub>1</sub>;... [ID;Link]<sub>n</sub>); Out([ID;Link]<sub>1</sub>;... [ID;Link]<sub>n</sub>)}

Здесь блок **Index** предназначен для хранения атрибутов, позволяющих позиционировать элемент в ЕИП-(ID), определять его принадлежность определенному классу-(idC), структуре-(idS), модели-(idC), слою-(idL). В блоке **View** содержится информация, определяющая вид элемента на экране монитора и информация о допустимых действиях с этим элементом. Блок **Lang** состоит из перечня названий (расширенных имен) элемента на различных языках. Группы индексов в блоках **Struct**, **mModel**, **iModel**, отвечают за положение элемента в соответствующих структурах и моделях, (рис. 1).

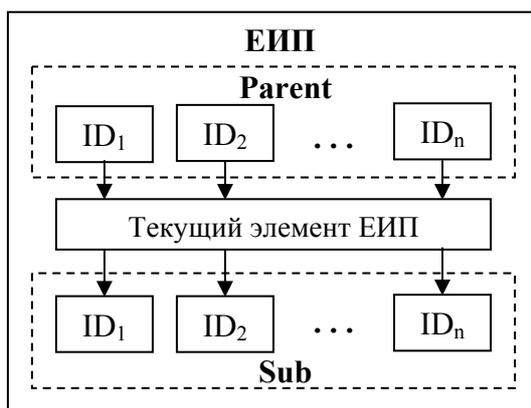


Рис. 1. Схема позиционирования текущего элемента ЕИП в структурах, моделях, слоях

По сути, последние блоки атрибутов содержат информацию о “главных” и “подчиненных” элементах текущего элемента ЕИП, а также информацию, позволяющую управлять состоянием “подчиненных” элементов в течение сеанса

работы пользователя, используя компоненты **iV** и **Mask** в блоке **mModel**. Под состоянием подразумевается характеристики элемента, определяющая его поведение, допустимые по отношению к нему действия, в частности, определять видимость элемента, изменять и удалять, копировать и клонировать элемент, выдавать права на конкретный элемент конкретному пользователю или группе, понимая, что каждый из них тоже имеет свой ИА и сам является элементом ЕИП. Маска состояний (**Mask**) элемента содержит набор состояний. Любому элементу допускается иметь свою маску состояний, обеспечивая тем самым свою оригинальную схему управления без ущерба для общей логики обработки масок состояний.

Индексы объединенные в группу **Controller**, соответствуют “головам” структур, описывающих сценарии поведения элементов ЕИП. Например, они могут отражать схемы управляющие последовательностью вызова сервисных функций и проектных процедур, что открывает в связи с этим новые возможности по формированию постановок задач оптимального проектирования проточных частей, их параметризации и т.п. Это по существу обеспечивает регламентирование выполнения процессов обработки данных (или цепочек таких процессов) и управление ими.

Данные блока **pModel** отражают схему связей внутренних параметров основных технологических и прикладных компонент системы либо параметров внешних источников с их аналогами-элементами ЕИП. Таким образом, используя значение индекса этого элемента и содержание компоненты **Link**, появляется принципиальная возможность организации обмена данными, как внутри ЕИП, так и с внешними источниками данных в операциях экспорта-импорта.

Например, одной из прикладных компонент ЕИП является проектная процедура реализующая задачу анализа многоступенчатой проточной части осевой турбины в одномерной постановке. Внутренний формат данных этой процедуры определяется десятью массивами, восемь из которых предназначены для входных данных, а два – для выходных. В этом случае блок **pModel** может иметь следующий вид:

```
pModel {In ([277;1;5];[321;1;6];[589;2;4];...[1542;8;17]);
          Out([511;9;4];[758;9;2];[451;9;7];...[977;10;25])}.
```

Здесь первое число в квадратных скобках соответствует индексу элемента-параметра в ЕИП, остальные два числа являются компонентами параметра **Link**, первая из которых соответствует номеру массива проектной процедуры, в котором должно находиться значение этого параметра, а вторая равна номеру ячейки массива, предназначенной для этого параметра. Следует отметить, что при создании схемы связей с внешними источниками данных, параметр **Link** может иметь и другую форму. Например, если в качестве внешнего источника данных взять файл сгенерированный программным комплексом AxStream, компании SoftInWay [4], который представляет собой XML-документ (фрагмент приведен на рис. 2), параметр **Link** целесообразно формировать из цепочки символов состоящей из двух идентификаторов – первый из них это имя Parent-тэга, а второй имя тэга-параметра. В этом случае блок **pModel** будет выглядеть следующим образом:

```
pModel{In([277;CStgClrn;d1];[321;CStgClrn;b1n];[589;CCasW;B1g];...[1542;CBladeW;z]);
          Out([511;CCylinder;Nc];[758;CCylinder;kpdi];[451;CCylinder;G];...[977;CCylinder;uCo])}.
```

Сформировав описанным выше способом схемы связей параметров всех проектных процедур, сервисных функций и внешних источников данных – мы

получаем принципиальную возможность обмена информацией как внутри ЕИП, так и в операциях экспорта-импорта данных с единым унифицированным механизмом обработки блоков **pModel** в не зависимости от используемых проектных процедур и форматов данных внешних источников.

```

. . . . .
<CCylinder>
  <Nc>53598.703545785262</Nc>
  <kpdi>0.8216522199848979</kpdi>
  <G>1.3688211355443358</G>
  <uCo>0.50728632041720301</u
. . . . .
<CStgClrn>
  <d1>0.004000000000000001</d1>
. . . . .
<CBladeW>
  <z>96</z>
. . . . .
<CStgFlow>
  <u1>109.95574287564274</u1>
  <c0>42.181968729961667</c0>
. . . . .
<CCasW>
  <B1g>32</B1g>
. . . . .
</ CCylinder >

```

Рис. 2. Фрагмент файла исходных данных в XML-формате

Необходимо также понимать, что в зависимости от класса элемента, набор блоков атрибутов может меняться. Так например, блок **Value** будет присутствовать в элементах класса “Параметр” и ”Массив”, блок **pModel** используется в элементах класса “Процедура” и т.д.

Инициализации атрибутов конкретного элемента ЕИП осуществляется через административную часть пользовательского интерфейса. В процессе “разворачивания” ЕИП в оперативной памяти компьютера адреса всех его элементов размещаются в соответствующих индексных словарях, где ключом для поиска элемента в ЕИП, структуре, модели, слое, сценарии и т.п. служит его уникальный индекс, а результатом поиска его в таком словаре будет адрес этого элемента в памяти компьютера. Учитывая, что алгоритмы поиска, использующиеся в индексированных словарях, на сегодняшний день являются одними из самых эффективных, мы получаем универсальный механизм поиска и адресации в ЕИП.

Очевидно также, что имея в составе атрибутов основных технологических и прикладных компонент блоки **pModel**, с помощью простой операции логического сложения можно “связать” параметры этих компонент напрямую между собой. Создав таким образом временный индексированный словарь, в котором в качестве ключа будет выступать цепочка символов **Link** одной из проектных процедур, а значением по этому ключу будет параметр **Link**, соответствующий другой проектной процедуре. В этом случае временный блок **pModel** будет иметь следующий вид:

```

pModel{In([1;5;CStgClrn;d1];[1;6;CStgClrn;b1n];[2;4;CCasW;B1g];...[8;17;CBladeW;z]);
Out([9;4;CCylinder;Nc];[9;2;CCylinder;kpdi];[9;7;CCylinder;G];...[10;25;CCylinder;uCo])}.

```

Таким образом, предлагаемая технология организации ЕИП с учетом возможностей по позиционированию его элементов в самом пространстве, структурах, моделях и слоях, а также универсальные возможности механизма обмена данными, основанного на формате блока **pModel**, позволяющего связывать соответствующие параметры-элементы ЕИП с внутренними параметрами элементов-процедур, с

параметрами внешних источников, а также организовывать прямые связи между внутренними параметрами различных элементов-процедур ЕИП, открывает новые и достаточно широкие возможности по регламентации и управлению вычислительными процессами в ЕИП. Для этих целей используется компонента пользовательского интерфейса «Диспетчер процессов», которая позволяет реализовать произвольные политики управления процессами, создавая соответствующие структуры сценариев процессов, как чисто иерархических, так и сетевых моделей этих процессов. В этом случае процесс оптимального проектирования проточной части турбины будет одним из сценариев вычислительного процесса в ЕИП. Примечательно, что звенья этого сценария по усмотрению пользователя могут меняться в зависимости от требуемой постановки задачи оптимизации, а также используемых проектных процедур и вычислительных алгоритмов.

Описанная выше технология информационного обмена в рамках ЕИП реализована в подсистеме “TurboOptProject” (ТОР) САПР «Турбоагрегат» с использованием библиотеки классов Qt и менеджера внешних источников данных ODBC.

Выводы:

- Логическая модель ЕИП базируется на едином реестре элементов этого пространства (ЕРЭ);
- в ЕРЭ регистрируются элементы, соответствующие предметным и размерным сущностям, технологическим и прикладным компонентам ЕИП САПР «Турбоагрегат». Атрибуты элементов содержат информацию об уникальном индексе элемента, его классе, о принадлежности элемента к структуре, модели, информационному слою, его состоянию, правах доступа и др.;
- уникальность индекса элемента и единое пространство индексов всех прикладных элементов в системе обеспечивает:
  - получение полной информации об элементе во время вычислительного процесса;
  - унифицированное управление сценариями вычислительных процессов;
  - унифицированное управление масками состояний и правилами переходов между состояниями элементов ЕИП;
  - унифицированное управление правами доступа к элементам;
  - общий механизм поддержки расширяемых атрибутов.

#### Литература

1. *Бойко А. В.* Разработка информационной среды и средств динамического управления информационными моделями данных сложных технических объектов применительно к САПР «Турбоагрегат» / А. В. Бойко, Ю. Н. Говорущенко, А. П. Усатый // Вестник Национального технического университета «Харьковский политехнический институт». Сборник научных работ. Тематический выпуск: «Энергетические и теплотехнические процессы и оборудование». – Харьков: НТУ «ХПИ». – 2006. – №5. – с. 36-42.

2. *Гайсарян С. С.* Объектно-ориентированные технологии проектирования прикладных программных систем // Центр Информационных Технологий <http://www.citforum.ru>.

3. *Кузнецов С.* Проблема качества баз данных для САПР // Открытые системы. – 2004. – № 2. <http://www.softiway.com>.

© Бойко А.В., Говорущенко Ю.Н., Усатый А.П., 2007