

Как следует из вышеизложенного, создается первая в стране компьютерная система визуализации (с помощью компьютерных программных средств) экологической экспертизы промышленных объектов. На основе этой системы будет создана соответствующая система для применения в практической деятельности эколога-эксперта.

ЛИТЕРАТУРА

1. Попов Э.В. Экспертные системы. Решение неформализованных задач в диалоге с ЭВМ.- М.:Наука, 1987.-283с.
2. Чепурных Н.В., Новоселов А.Л. Комплексная социо-эколого-экономическая экспертиза региона. Системный подход // Экологическая экспертиза №3.- М.: 1996. - С.3.
3. Сухорукова И.Г. Компьютерная обучающая система "Оценка воздействия на окружающую среду и экологическая экспертиза" // Материалы Республиканской научно-технической конференции студентов и аспирантов. - Гомель: ГГУ, 1998.- 49с.

УДК 519*714

Н. Н. Дорожкина, аспирант

ОБЪЕКТНО-ОРИЕНТИРОВАННЫЕ ТЕХНОЛОГИИ В АВТОМАТИЗАЦИИ ПРОИЗВОДСТВА

The paper presents a new approach to the usage of object-oriented technologies in manufacturing control. Quite a full representation of the object-based specifications of the manufacturing items is given with special attention to the problem specification within the technological plan class.

1. Введение

Разработка систем автоматизации производства эффективно реализуется на основе объектно-ориентированных технологий [1] программирования (проектирования). Эффективность подхода базируется на следующем:

1. Объектно-ориентированный подход представляет языковые средства для построения формализованных спецификаций систем.

2. Современные объектно-ориентированные языки позволяют (на их основе) использовать библиотеки классов для поддержки процессов создания и использования сложных программных систем.

Эти два фактора играют ключевую роль в подходе к реализации производственных систем автоматизации. В настоящей статье представлена детальная реализация объектного подхода к построению системы автоматизации производственного планирования.

2. Иерархия классов

Иерархическая структура классов для описания системы производственного планирования представлена на рис. 1.

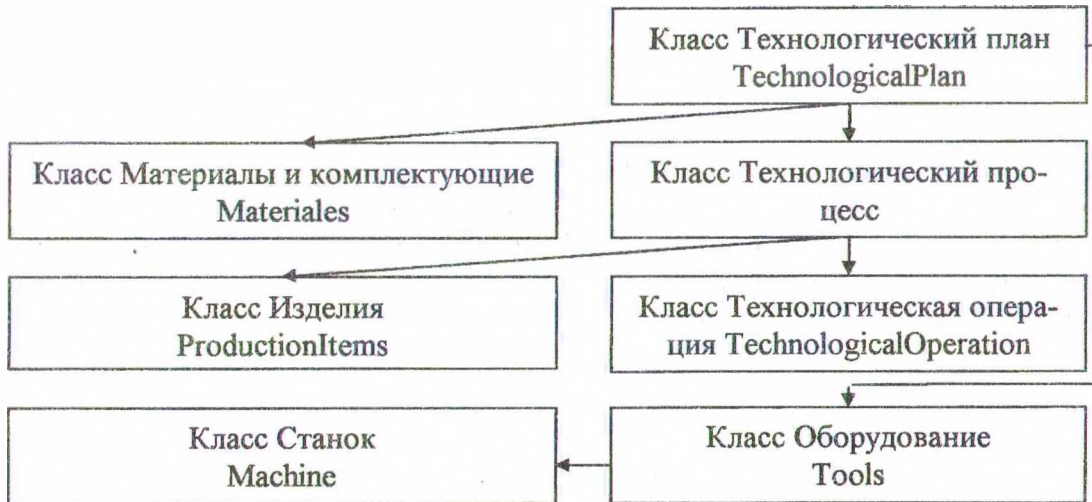


Рис. 1. Иерархия классов системы производственного планирования

2.1. Класс TechnologicalPlan

Приведем структуру класса и необходимое пояснение для его переменных, методов и событий.

Переменные:

- TechnologicalProcessList – список технологических процессов;
- MaterialList – список материальных резервов (запасов);
- ToolList – список оборудования по шагам технологического процесса;
- ToolProductionRates – производительность оборудования;
- PlanList – план выпуска продукции;
- CurrentOutList – список объемов изготовленных изделий;
- OrderList – порядок запуска изделий (групп) на обработку;
- RestoredList – список восстанавливаемого оборудования;
- TimeList – временное расписание запуска изделий на обработку;
- OperTimeList – операционные времена обработки изделий на операциях;
- SystemClock – системные часы;
- CostList – список стоимостей готовых изделий;

ToolCostList – стоимость использования оборудования (ед. времени работы);

MaterialCostList – список стоимостей единиц материала;

DueDateList – список директивных (плановых) времен выполнения каждого наименования продукции;

TechnologicalMatrixList – матрица технологических затрат (Леонычева); **ProductionFunction** – производственная функция в форме полинома;

PlanningProblemSpecification – спецификация производственной задачи;

ProductionFactorsDatabase – база данных производства;

TechnologicalGraph – технологический граф;

RepairingCosts – список стоимостей единицы времени ремонта оборудования;

RepairedLaunchingTimes – времена ввода отремонтированного оборудования.

Методы:

Add – добавление элемента; **Insert** – вставка элемента;

Delete – удаление элемента; **Clear** – очистка списка;

Destroy – уничтожение экземпляра;

SolveProblem – решить производственную задачу;

CompileProblem – скомпилировать задачу;

EvaluateFactors – оценить производственные факторы;

CreateProductionFunction – построить производственную функцию;

EditProductionFunction – редактировать производственную функцию;

EditProductionProblem – редактировать производственную задачу;

Simulate – запустить и моделировать производственный процесс.

События:

OnCreateP – создание компонента; **OnDeleteP** – удаление компонента;

OnClockReset – сброс системных часов;

OnSimulationReady – готовность данных для моделирования;

OnSimulationFinish – завершение моделирования;

OnCompileError – ошибка компиляции;

OnProblemFinsh – завершение расчета.

Класс Технологический план является базовым (исходным) классом рассматриваемой иерархии. Он содержит данные для описания технологической системы в целом. В него введены две важнейшие системные переменные: **ProductionFunction** и **PlanningProblemSpecification**.

Обе переменные строковые. Производственная функция представляется компонентом общего вида

$$A_1 * X_1^{\alpha_1} + A_2 * X_2^{\alpha_2} + \dots + A_n * X_n^{\alpha_n},$$

где A_i – вещественные коэффициенты; α_i – вещественные показатели степени; X_i – производные факторы (соответствуют именам столбцов (номерам столбцов) базы данных производства ProductionFactorsDatabase).

Подробно остановимся на поле PlanningProblemSpecification. В этом поле записывается математическая постановка задачи производственного планирования [2]. Для записи задачи производственного планирования используется язык спецификаций. Компиляция спецификаций определяет, верно записана задача или нет. В первом случае задача может быть отправлена на выполнение. Система допускает следующий перечень задач:

- решение систем неравенств в целых и вещественных числах;
- решение систем дизъюнктивных неравенств в вещественных числах;
- решение задачи линейной оптимизации;
- решение задачи оптимизации для системы с дизъюнктивными неравенствами в вещественных числах.

Для записи задач используется язык спецификаций.

Рассмотрим пример формализации постановки:

PROBLEM Z1

VAR

X [1..100] : format;

Y [1..100] : format;

B [1..100] : format;

C [1..100] : format;

function F

B × X → max

Restrictions

C × X ≥ G

Where

ASSOCIATE B with spec1

ENDASSOCIATE


```

ASSOCIATE C with spec2
ENDASSOCIATE
ASSOCIATE Y with spec3
ENDASSOCIATE
ENDPROBLEM

```

В разделе VAR объявляются переменные и объекты. Спецификатор `format` определяет тип объекта и принимает значения `integer`, `real`, `boolean`, `compoundinteger`, `compoundreal`, `compoundboolean`.

Тип `compound...` используется для записи дизъюнктивных ограничений. Функции и ограничения могут быть записаны в общем (векторно-матричном) виде (как выше) либо в виде конкретных выражений:

$$1x.1 + 3x.2 + 4x.3 \geq -1 \text{ (и т.д.)},$$

где индикатор $x.i$ есть i -й элемент списка X .

Аналогично $x.i,j$ есть (i,j) -й элемент массива X .

Спецификаторы `spec1`, `spec2`, `spec3` могут принимать следующие возможные значения: `RandomObject(100)` (случайное число в диапазоне от 0 до 100); `MaterialCostList` (переменная “массив” связывается со списком `MaterialCostList`, определяющим поле компонента `TechnologicalPlan`).

File “C:\word.dat” - данные в список заносятся из файла.

Построенная проблема передается на решение в метод `SolveProblem` следующим образом: `TechnologicalPlan1.SolveProblem (Z1)`.

Решение задачи приводит к формированию значений неизвестных переменных. Эти значения должны быть ассоциированы с публикуемыми свойствами компонента `TechnologicalPlan` с тем, чтобы получить результаты расчетов.

Подробнее остановимся на спецификаторе `compound`, позволяющем записывать дизъюнктивные неравенства.

Рассмотрим следующий пример:

$$\left\{ \begin{array}{l} 2x_1 - x_2 + 3x_3 \rightarrow \max \\ x_1 - x_2 + 4x_3 \geq 2 \vee -x_1 - x_2 \geq -1 \\ -x_1 + x_3 \geq 0 \vee x_2 \geq 0 \vee x_3 \geq 1 \\ x_1 + x_2 - 2x_3 \geq 1 \\ x_1 \geq 0 \\ x_2 \geq 0 \\ x_3 \geq 0 \end{array} \right. \quad (*)$$

Матрица коэффициентов C типа compound имеет следующую структуру:

$$C = \left[\begin{array}{ccccccccc} 1, & -1, & 4, & -1, & -1, & 0 & & & \\ -1, & 0, & 1, & 0, & 1, & 0, & 0, & 0, & 1 \\ 1, & 1, & -2 & & & & & & \end{array} \right]$$

Символ «;» определяет простые неравенства в дизъюнктивных. Поскольку число простых неравенств в дизъюнктивных заранее не фиксировано, массив C кодируется списком

$$C = [1, -1, -4; -1, -1, 0 / -1, 0, 1; 0, 1, 0; 0, 0, 1 / 1, 1, -2].$$

Слеш («/») отделяет одно дизъюнктивное неравенство от другого. Список C объявляется следующим образом:

VAR C : compoundinteger;

Запись $C * X \geq B$, где $X = \{x_1, x_2, x_3\}$ и $B = [2; -1 / 0, 0, 1 / 1]$

– суть также список типа compoundinteger, может быть очевидным образом преобразована в подстановку задачи (*).

Таким образом, компонент TechnologicalPlan позволяет формализовать широкий класс задач производственного планирования, который можно формализовать в рамках задач с линейными простыми и дизъюнктивными неравенствами.

2.2. Класс TechnologicalProcess

Этот класс позволяет описывать отдельно взятый технологический процесс. Он имеет следующие свойства:

Name – имя (номер) технологического процесса;

Count – число всех операций технологического процесса;

OSur – текущая операция технологического процесса;

ToolList – список оборудования по операциям;
 MaterialNormsList – список норм расхода материала по операциям;
 CondList – список условий перехода на следующую операцию;
 PlanList – план выпуска изделий по наименованиям;
 DefectAmount – норма брака;
 OperationGraph – граф предшествования технологических операций.

Методы:

AddP – добавить элемент; InsertP – вставить элемент;
 DeleteP – удалить элемент; ClearP – очистить список;
 Evaluate – оценить истинность условий CondList;
 FindReady – построить список готовых к выполнению операций.

События:

Onempty – завершение обработки последнего изделия;
 OnStart – запуск технологического процесса на выполнение;
 OnUnexpected – возникновение незапланированного события
 (например, нехватка материала, сбой и др.);

OnFinish – завершение технологического процесса.

Представленный класс является интерфейсом между классом TechnologicalPlan и TechnologicalOperation.

2.3. Класс TechnologicalOperation

Данный класс служит для описания технологических операций и является одним из важнейших классов в иерархической структуре классов. Этот класс содержит следующие переменные:

Name – имя (номер) операции; Defect – вероятность брака;
 Time – время выполнения операции (трудоемкость);
 PredecessorsList – список непосредственных предшественников;
 SuccessorsList – список непосредственных преемников;
 MaterialList – список запасов материалов и комплектующих;
 Priority – приоритет операции;
 CurrentState – текущее состояние операции;
 StartTime – время запуска операции;
 DueTime – директивное время запуска операции;
 WaitFareFunction – функция штрафа за ожидание начала выполнения операции;

LateFareFunction – функция штрафа за нарушение директивного времени; TotalProcessed – общее число обработанных изделий;

CurrentProcess – технологический процесс;
 OrdinalNumber – порядковый номер операции;

PlanList – план выпуска изделий.

Методы:

Add, Insert, Delete, Clear – общие;

FindexpectedFinishTime – возвращает ожидаемое время завершения операции;

ExecFairFunction – производит вычисление функции штрафа за превышение директивного времени;

ExecWaitFunction – производит вычисление функции штрафа за время ожидания операции;

SetPriority – устанавливает новое значение приоритета.

События:

OnStart – запуск операции; OnFinish – завершение операции;

OnExceedDueTime – превышение директивного времени;

OnMaterialExhaust – нехватка материала;

OnDefect – появление брака;

OnDisabilityofOperation – невозможность выполнения операции.

2.4. Класс Tools

Свойства:

State – состояние; ProductionSpeed – производительность;

OperatingDuration – время наработки на отказ;

TechologicalProcessesList – список технологических процессов, где используется оборудование;

PreConditions – формула предусловий для запуска операции;

PostConditions – формула постусловий завершения операции;

CostPerTimeOper – стоимость единицы времени работы;

CostPerTimeRepair – стоимость единицы времени ремонта;

BufferSize – размер приемного буфера.

Методы:

Add, Insert, Delete, Clear – общие;

EvaluatePreconditions – оценить предусловия;

SetPostConditions – установить постусловия.

События:

OnCreate – создание элемента; OnDelete – удаление элемента;

PreConditionsSettled – предусловия выполнены;

PostConditionsSettled – постусловия установлены;

PreConditionsBlocked – предусловия заблокированы.

2.5. Класс ProductionItems

Свойства:

Number – номер; GroupNumber – номер группы;

Machine – нахождение; TechnologicalProcess – техпроцесс;

TechnologicalOperation – операция;

State – состояние (покоится до операции, обработка, покоится после операции);

FinishTime – время завершения операции; Priority – приоритет;

NextOperation – номер следующей операции;

NextStartTime – время начала следующей операции;

ArriveTime – время нахождения на текущей операции;

TotalProcessingTime – время нахождения в системе;

TotalWaitingTime – накопленное время ожидания;

OperationList – список операций технологического процесса;

DefectAvailable – наличие дефекта.

Методы:

Add, Insert, Delete, Clear – общие;

GetOperation – получить операцию по номеру из системы;

MoveToNextOperation – перевести на следующую операцию;

Terminate – уничтожить изделие; Create – создать изделие;

StartOperation – запустить изделие на обработку на текущем станке;

StopOperation – остановить операцию на текущем станке;

PutIntoBuffer – поместить изделие в буфер;

GetFromBuffer – вставить изделие из буфера.

События:

OnCreate – создание элемента;

OnDelete (Terminate) – удаление элемента;

OnOperationStart – запуск операции;

OnOperationFinish – завершение операции.

3. Реализация

Рассмотренные классы ориентированы на реализацию в форме визуальных компонентов пользователя (например, системы Delphi 5.0). Уже один компонент TechnologicalPlan представляет набор широких возможностей по решению задач планирования производственного процесса. Реализация системы в полном объеме позволяет построить полностью адекватную модель системы управления производственным процессом на уровне цеха или линии. Пользователь должен

обеспечить ввод исходных данных и создание программного интерфейса. План первой очереди реализации системы включает следующие действия: создание класса `TechnologicalPlan` и библиотеки математических методов, поддерживающих решение через метод `SolveProblem`.

ЛИТЕРАТУРА

1. Основы современных компьютерных технологий / Под ред. А.Д. Хомоненко. – СПб, 1998. – 448 с.
2. Герман О.В., Дорожкина Н.Н. Об одной общей задаче производственного планирования // Труды Белорусского государственного технического университета. Выпуск VII. -Минск, 1999. - С. 29.

УДК 681.3.07

С. Б. Макась, аспирант

КЛАССИФИКАЦИЯ МЕТОДОВ ВВЕДЕНИЯ АППАРАТНОЙ ИЗБЫТОЧНОСТИ В КРИСТАЛЛЫ ДИНАМИЧЕСКИХ ОПЕРАТИВНЫХ ЗАПОМИНАЮЩИХ УСТРОЙСТВ (ДОЗУ)

Generalization and classification of the hardware redundancy implantation in DRAM are presented.

1. Причины возникновения отказов в ДОЗУ

Важнейшее направление совершенствования ДОЗУ связано с увеличением объема хранимой информации и уменьшением геометрических размеров элементов. Это обусловлено увеличением степени интеграции элементов на кристалле, что приводит, с другой стороны, к снижению надежности хранения информации вследствие влияния на кристалл различных дестабилизирующих факторов.

Известны основные виды дестабилизирующих факторов и вызванные ими нарушения работоспособности: сбои и отказы. Отказ – устойчивое нарушение работоспособности. Сбой - нарушение работоспособности, устраняемое без проведения ремонта. Искажение хранимой или считанной в ДОЗУ информации под воздействием отказа или сбоя называют ошибкой.

Анализ показал, что известные дестабилизирующие факторы и их последствия могут быть классифицированы следующим образом:

1) Характерные для запоминающих элементов (ЗЭ). К ним относятся: