

МЕТОД РЕАЛИЗАЦИИ БЛОКА ТРАНЗАКЦИОННЫХ ДАННЫХ РЕЛЯЦИОННОЙ СУБД для ОПИСАНИЯ ПРОИЗВОДСТВЕННОГО ОБОРУДОВАНИЯ

И.С. Решетников, П.Н. Коснырев (ООО "Газпром центрремонт")

Рассматривается метод организации физической структуры данных в реляционной СУБД, основанный на рекомендациях стандарта ISA-95 и метамодели языка UML. Предлагаемая модель функционально ориентирована, прежде всего, на описание структуры сложносоставного технологического оборудования.

Ключевые слова: ISA-95, метамодель, класс, атрибут, объект, отношение.

Введение

При построении корпоративных информационно-вычислительных систем, имеющих дело с описанием сложных по структуре производственных комплексов, одной из основных задач является построение системы надежного и гибкого хранения данных об оборудовании.

Задача описания оборудования не относится к категории широко обсуждаемых в литературе, в отличие от проблем ведения НСИ, БД РВ и т.п. Связано это с тем, что на большинстве предприятий вопросы реализации связи между производственными заданиями и характеристиками оборудования пока не поднимается. Перечень оборудования если и требуется, то только для корректного отнесения затрат ТООиР и ведения соответствующих технологических карт. В этом случае описание структуры оборудования ведется внутри ERP-систем на встроенных механизмах (например, при помощи "технических мест" и "единиц оборудования" в SAP ERP). При этом набор параметров минимален, специальных требований к целостности не предъявляется. Технологическая структура единиц оборудования и промплощадок не ведется вовсе (плоский список) или ведется в простой иерархической структуре (например, оборудование привязано только к цеху).

Более детально оборудование иногда описывается в системах оценки надежности и технического состояния (или оптимизации режимов работы). Но даже в случае наличия на предприятии таких систем, данные хранятся в специализированной БД, часто локальной. При этом накопление и учет данных привязывается не к реальной структуре оборудования, а к модельной, которая всегда является неким компромиссным упрощением.

Очевидно также, что данные по составу и характеристикам оборудования, по его доступности, точности выполнения тех или иных операций являются слишком "непостоянной" информацией и не могут быть отнесены к разряду НСИ.

Если на крупном предприятии решается комплексная задача по автоматизации его деятельности, то сведения о структуре оборудования и об основных текущих технологических показателях востребованы сразу в нескольких процессах (ниже приведен далеко не полный перечень):

- оценка готовности производственного участка к выпуску конкретной продукции;

- планирование загрузки оборудования на уровне производственного участка;
- единая точка сбора и контроля эксплуатационных показателей;
- оценка эффективности использования оборудования;
- контроль соответствия заявленных показателей обработки допустимым (контроль качества);
- перспективное и оперативное планирование работ ТООиР;
- оценка целостности комплекса технологического оборудования (необходимые элементы защиты, резервные подключения, и т.д.);
- оценка технического состояния и надежности парка оборудования;
- контроль мероприятий по обеспечению надежности;
- обеспечение безопасности работ на промплощадке;
- формирование программы технического перевооружения.

Даже поверхностный анализ хранимых данных показывает, что многие задачи пересекаются по учитываемым показателям, некоторые требуют ретроспективного хранения данных, другие сами порождают показатели, которые должны в будущем учитываться и т.д. Поэтому, чтобы избежать разрозненного хранения данных по программным комплексам (хранение данных на разных уровнях организационной иерархии допускается), система описания технологических объектов должна отвечать нескольким простым требованиям:

- уметь описывать 100% оборудования и показателей, учитывая 100% функциональных взаимосвязей между его элементами;
- прозрачно учитывать технологическую иерархию установленного оборудования;
- иметь внутренние механизмы контроля "базовой" целостности (по кратности подключений, по принадлежности и т.д.);
- иметь встроенные инструменты контроля входных/выходных данных;
- хранить ретроспективные данные;
- быть основой для построения типовых схем и систем сбора необходимых данных;
- иметь открытый интерфейс для работы с внешними системами;
- иметь гибкую, подстраиваемую под задачи, систему регистрируемых показателей.

Исходя из предъявляемых требований, накладываются и ограничения на возможные подходы к реализации данной задачи. В существующих системах БД технологического оборудования строятся на основе современных реляционных СУБД (РСУБД), и блок транзакционных данных представляет собой нормализованную структуру, исключающую избыточность и обеспечивающую целостность данных на уровне встроенных механизмов СУБД. Однако нормализация структуры не всегда обеспечивает нужный уровень связности данных, так как не всегда с ее помощью возможно описать системы со сложной внутренней иерархией при обеспечении необходимого уровня согласованности. Вторым существенным недостатком нормализации является сложность внесения корректировок в структуру данных, необходимость которых возникает при изменении модели описания единиц оборудования.

Задача описания структуры оборудования для целей оперативного управления производством не может быть решена и системами класса PLM/PDM [1]. Для решения поставленных задач недостаточно описания лишь статической электронной модели объекта, а требуется динамическое описание комплекса сложных объектов с динамической внутренней структурой при наличии редких структурных перестроек с условием хранения истории изменений и обеспечения целостности. Такое дополнительное условие делает неприменимым использование стандартных технологий вроде CALS или построения исполнительной документации с помощью подхода "Как построено". Применимость данного вида технологий ограничивается разовым описанием системы при дальнейшем внесении несущественных изменений, не затрагивающих структуру объекта, при этом за целостность отвечает, как правило, оператор, осуществляющий ввод данных.

Для практического применения важен гибкий подход к описанию структуры оборудования и его параметров. Рекомендации по реализации такой модели содержатся в стандарте ANSI/ISA-95.00.01-2000 [2]. Предлагаемая структура модели описания оборудования представлена на рис. 1.

Рассмотрим базовые предпосылки к использованию в качестве базовой такой "зацикленной" схемы организации данных. Построение структуры данных предполагает приписывание каждой сущности предметной области некоторого фиксированного домена (типа, области значений) [3]. С другой стороны, обычная практика использования транзакционных систем явно или не явно предполагает нарушение этого принципа, поскольку образ предметной области меняется со временем, что приводит к модификации структуры данных. При использовании сильно нормализованной реляционной структуры (что характерно для транзакционных систем), утрачивается одно из основных ее преимуществ — простота и концептуальная ясность. Кроме этого, изменение струк-

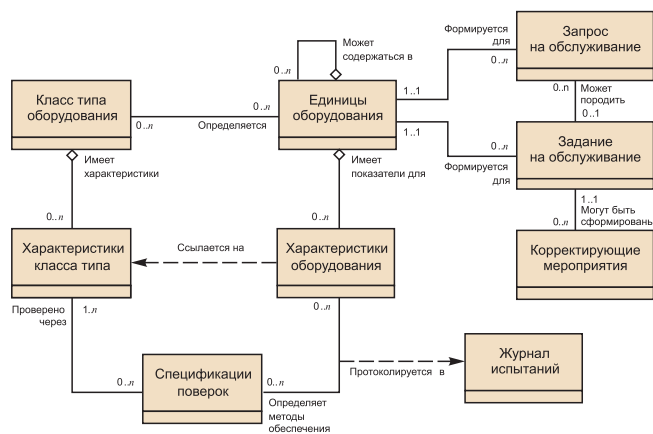


Рис. 1. Модель описания оборудования в стандарте ISA-95

туры данных связано со сложностями сопровождения и администрирования РСУБД, что особенно критично в случае распределенных систем.

Естественным выходом из указанной ситуации является использование единой реляционной схемы для представления произвольных прикладных данных, модель которых неким образом специфицирована и хранится в реляционной схеме в виде фиксированным образом структурированных метаданных.

Именно такой подход описан в статье, и далее под логической структурой данных будем понимать спецификацию модели прикладных данных, а под физической структурой данных — непосредственно реляционную схему. В работе рассматривается практический подход к организации транзакционных данных в РСУБД на примере СУБД Oracle 10g, обеспечивающий возможность расширять логическую структуру данных в процессе функционирования информационной системы и отвечающий основным требованиям к системам описания технологических комплексов. В статье приведено описание конкретной реализованной системы и отмечены те детали, которые авторы считают особенно важными, и которые могут рассматриваться как рекомендации при реализации подобных систем в рамках других проектов.

Реализация системы описания технологического оборудования

Поскольку стандарт ISA-95 не дает конкретных рекомендаций по практической реализации модели оборудования, подход к разработке логической структуры, описываемый в работе, базируется на метамодели языка UML [4], а точнее той ее части, которая относится к диаграмме классов. Так, все объекты предметной области группируются в относительно устойчивые и независимые совокупности — классы, для каждого из которых определяется набор именованных свойств — атрибутов. Класс (потомок) может быть создан на основе уже существующего (родительского) класса, при этом потомок наследует атрибуты родителя. Связи между объектами предметной области описываются отношениями между классами, среди которых выделяются: ассоциация (общий случай

отношения), агрегирование (отношение вида "часть-целое") и композиция (агрегирование с четко выраженным отношением владения, когда время жизни частей и целого совпадают). Отношения двух последних видов являются транзитивными и асимметричными. Структурная часть отношения, определяющая какой класс чувствует в отношении и какую роль играет, называется полюсом отношения. Для каждого полюса отношения может быть задана кратность, определяющая допустимое число объектов, связываемых экземплярами соответствующего отношения. Кратность задается целыми неотрицательными числами и указывается в виде точного значения (например, 1) или интервала значений (например, 0..7). Если интервал значений не ограничен сверху, кратность задается выражением вида 0..*.

При описании больших систем с потенциально большим числом классов возникает необходимость организовывать эти сущности в более крупные блоки. Для организации моделируемых сущностей в группы, по аналогии с семантикой языка UML, используются пакеты, которые содержат другие классы и пакеты. Помимо группировки сущностей пакеты также определяют границы видимости. Видимость класса в пакете указывает, может ли он иметь отношения с классами, находящимися в других пакетах. Каждый элемент (класс, пакет) может принадлежать только одному пакету.

Реализация предложенной концепции формирования метаданных позволяет создавать сложные логические структуры для описания модели предметной области, состоящей из большого числа различных объектов, с формализацией ограничений и обеспечением целостности данных. Для практической реализации объектной модели была применена схемонезависимая стратегия отображения информационной схемы [5], физическая модель данных представ-

лена на рис. 2. Для упрощения некоторые вспомогательные таблицы, в том числе блок хранения ретроспективных данных, журналы изменений и т.д. на схеме опущены.

Все таблицы в БД можно разделить на две категории: таблицы для хранения метаданных — описания пакетов (PACKAGE), классов (CLAZZ), атрибутов (ATTRIBUTE) и отношений, и таблицы для хранения собственно данных — объекты (OBJECT) и их характеристики (ATTRIBUTE_VALUE). В качестве значений для первичных ключей применяется универсальный 16-байтный идентификатор UUID (Universally Unique Identifier консорциума OSF). Его можно считать уникальным с приемлемым уровнем вероятности.

Для обеспечения контроля качества данных на уровне модели в таблице классов в качестве параметра хранится число допустимых экземпляров (или кратность) классов. Простым заданием ссылки на объект моделируется ассоциация "один ко многим". Полюсу ассоциации, к которому принадлежит класс с атрибутом-ссылкой, соответствует кратность 0..*. Указанный интервал кратности можно ограничить, задав верхний предел. Кратность, соответствующая противоположному полюсу ассоциации, равна 0..1. Ассоциации "многие ко многим" всегда моделируются с помощью класса-ассоциации.

Исходя из опыта проектирования прикладного ПО, можно говорить о том, что подавляющее большинство описываемых объектов предметной области должно иметь некое интуитивно понятное наименование (для обеспечения целостности данных это не важно). Для ряда объектов название задается явно при их создании или редактировании. Однако для некоторых объектов наименование удобнее формировать из значений их атрибутов по определенному шаблону, например: для некоторого узла в составе сборочной единицы название можно формировать как название сборочной единицы плюс номер узла по спецификации к технологической схеме. Шаблон для автоматического формирования наименования объекта хранится в описании класса. Так для примера, описанного выше, шаблон может выглядеть как "%1, узел №%2", где %1 и %2 — порядковые номера атрибутов в названии объекта.

Еще одной особенностью предлагаемой модели, направленной на практическую реализацию, является работа с типами данных. Всего в системе определено восемь типов данных: число, строка, текстовое поле, булев тип, год, дата, время и дата-время. Для каждого типа в таблице описания типов (DATA_TYPE) имеется запись, которая содержит формат типа. Формат представляет собой PL/SQL код, преобразующий входные данные к виду, в котором они будут храниться в БД. Например, для типа "год" формат будет иметь следующий вид: `to_char(to_date(decode(length(trim('%s')),4,'01.01.'||'%s','%s'),'dd.mm.yyyy'),'yyyy')`. При редактировании данных в системе все вхождения %s в строке формата автоматически заменяются на новое значение атрибута, после чего полученный код испол-

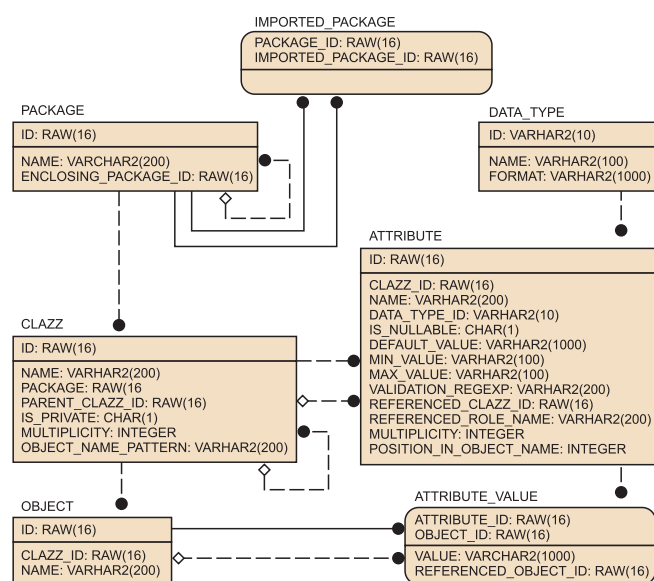


Рис. 2. Диаграмма физической структуры данных в нотации IDEF1X

няется, а результат сохраняется в БД. Встроенный код может быть достаточно сложным и включать вызовы пользовательских функций, что позволяет строить унифицированные простые в реализации инструменты ввода данных.

Все значения атрибутов простых типов хранятся в поле типа VARCHAR2, значения атрибутов-ссылок заносятся в ссылочное поле, которое является внешним ключом к таблице объектов.

С целью обеспечения безопасности и целостности информации, логика работы с данными, хранящимися в описанной структуре, инкапсулирована в слое хранимых процедур [6]. Прямая вставка и изменение данных в таблицах запрещена. Эти процедуры обеспечивают соблюдение требований целостности информации, и в случае обнаружения ошибки в получаемых данных инициируют соответствующее программное исключение. Процедуры собраны в одном PL/SQL пакете. Так как состояние PL/SQL пакета в течение сессии сохраняется, идентификатор текущей сессии (пакета, класса, атрибута и т. д.) хранится в соответствующей переменной PL/SQL пакета.

Генерация первичных ключей (UUID) вынесена в отдельную функцию, которая работает через вызов системной функции SYS_GUID() и при необходимости может быть переопределена, что позволяет гибко варьировать метод формирования идентификаторов.

Управление сущностями выполнено через специализированные процедуры, каждая из которых отвечает за элементарную операцию создания, удаления, изменения сущности с соблюдением определенных требований. Так, для гарантии непротиворечивости информации при работе с пакетами требуется обеспечить выполнение следующих условий:

- уникальности имен сущностей (пакетов и классов) внутри объемлющего пакета;
- отсутствия рекурсивной вложенности пакетов;
- соблюдения границ видимости в соответствии с установленными правилами;
- каскадного удаления всех сущностей, содержащихся в удаляемом пакете.

Процедуры для работы с классами гарантируют соблюдение следующих ограничений:

- имена классов внутри пакета уникальны;
- при изменении шаблона для автоматического формирования названий объектов автоматически обновляются названий всех экземпляров класса и всех значений атрибутов-ссылок, указывающих на экземпляры данного класса;
- при объявлении класса "закрытым" внутри пакета или при перемещении класса из одного пакета в другой, если данный класс имеет отношения с другими классами, необходимо соблюдать условия видимости в соответствии с правилами;
- отношения обобщения не должны быть рекурсивными;
- необходимо контролировать соответствие значения кратности класса числу его экземпляров;

- при удалении класса должно происходить каскадное удаление его экземпляров, атрибутов (включая значения) и отношений, которыми связан данный класс с другими классами;

Аналогично устроен блок процедур для работы с атрибутами, отвечающий за их создание, задание названия, свойств, шаблонов проверки корректности данных (через regexr), вхождение значений атрибутов в автоматически формируемое название объекта. При работе с атрибутами требованиями обеспечения целостности данных являются:

- контроль соответствия значений атрибутов задаваемым ограничениям;
- соответствие вхождения значений атрибутов в формируемое автоматически название объекта заданному порядку;
- каскадное удаление значений атрибута при удалении самого атрибута.

В отдельную группу вынесены процедуры для работы с ассоциациями. Любое отношение ассоциации опеределяется двумя полюсами. Для каждого полюса ассоциации должен быть указан класс, роль данного класса в ассоциации и кратность. Для создания ассоциации необходимо передать в качестве входных параметров заполненные соответствующим образом записи для полюсов данной ассоциации. В зависимости от значений кратности в полюсах ассоциации будет создан либо атрибут-ссылка для соответствующего класса (отношение "один ко многим"), либо класс-ассоциация (отношение "многие ко многим"). Названия экземпляров класса-ассоциации автоматически формируются из значений атрибутов-ссылок, соответствующих ее полюсам.

Для обеспечения целостности данных при работе с ассоциациями необходимо учитывать следующее:

- связи, описываемые ассоциацией, должны соответствовать значениям кратности, определенным для ее полюсов;
- отношения агрегирования должны отвечать требованиям асимметричности (отсутствие рекурсии);
- для класса-ассоциации должны быть определены два атрибута-ссылки, указывающих на ее полюса.

При работе с объектами условия обеспечения целостности таковы:

- число объектов не должно противоречить значению кратности соответствующего класса;
- при создании объекта необходимо сформировать значения атрибутов, для которых определено значение по умолчанию;
- при удалении объекта-композиата необходимо удалить все объекты, входящие в его состав;
- при изменении названия объекта, необходимо привести в соответствие автоматически формируемые названия, имеющие вхождения атрибута-ссылки, указывающего на заданный объект.

При работе со значениями атрибутов объектов в соответствующих функциях и процедурах учтено, что:

- значение должно удовлетворять ограничениям, заданным для соответствующего атрибута;

- значение атрибута-ссылки не должно противоречить кратности, указанной для соответствующей ассоциации;

- пустому значению атрибута соответствует отсутствие записи в таблице значений атрибутов.

Не все действия, описываемые приведенными процедурами, являются атомарными. Здесь под атомарностью понимается, что операция является наименьшим, неделимым блоком алгоритма изменения данных. Так, например, мы не можем создать только объект, для которого определен атрибут, не допускающий пустых значений. В данном случае атомарной операцией будет являться создание объекта и задание значения указанного атрибута. В связи с этим проверку некоторых ограничений необходимо откладывать до завершения выполнения определенного блока процедур. В PL/SQL пакете предусмотрены процедуры для объявления начала и окончания атомарного блока. Вызов процедур проверки ограничений для каждой операций, выполняемой внутри блока, ставится в очередь и производится только при вызове процедуры окончания блока.

Реализованная на базе приведенных выше принципов система обладает всеми необходимыми свойствами для описания сложного по внутренней структуре технологического оборудования крупного предприятия. При этом данные, хранящиеся в системе, могут быть доступны (с помощью встроенных механизмов СУБД) для внешних систем в удобной форме, в том числе развернуты в псевдо-нормализованную структуру на основе представлений. При помощи несложных триггеров типа INSTEAD OF в этой структуре может быть реализована и полноценная работа с данными в рамках отдельного программного комплекса.

Поскольку все манипуляции с данными проходят через единую точку контроля (PL/SQL-ный программный блок), целостность и защита данных обеспечивается на нужном уровне. Каждый программный комплекс "видит" свои показатели, в то время как общий состав характеристик может претерпевать существенные изменения. Введение "виртуальных" характеристик, которые по определенным правилам, примерно как наименование объекта, формируются при корректировке данных, решает проблемы наследственности "снизу вверх" и использования разных внешних потребителей информации.

Заключение

В работе рассмотрена методика универсальной (то есть не привязанной к определенной предметной области) организации структуры данных в реляционной СУБД, основанная на метамодели языка UML с учетом рекомендаций стандарта ISA-95. Указанный подход может быть полезен при решении широкого спектра прикладных задач, связанных с описанием сложного по структуре оборудования и со сбором первичной информации в блок транзакционных дан-

ных. Анализ сформированных метаданных позволяет упростить и решение ряда прикладных задач при разработке специализированного ПО, в частности:

- разработку CASE средств для формирования модели предметной области;

- наложение сложных структурных ограничений на модель предметной области;

- автоматизацию формирования форм ввода и отображения информации при разработке типовых программных приложений для работы с данными;

- организацию синхронизации данных в распределенных, в том числе иерархических системах с переменным набором описываемых объектов и их характеристик;

- ведение журналов изменения как самих данных, так и их структуры;

- ведение многомерного хранилища данных.

Использованные в предлагаемом решении подходы позволяют без дополнительных усилий сделать систему описания иерархически-распределенной. Благодаря уникальности идентификаторов описание объекта на верхнем уровне иерархической информационной системы будет представлять собой композицию всех субмоделей, построенных на основе базовой структуры в нижележащих уровнях иерархии. Это существенным образом облегчает построение распределенных структур хранения данных и повышает уровень их согласованности и непротиворечивости, так как все ограничения заложены на уровне единой объектной модели.

Данный подход наиболее эффективен, когда набор регистрируемых параметров относительно невелик (десятки-сотни для каждого типа объекта), но число объектов велико и требуется возможность динамического изменения структуры описываемых объектов при гарантированном сохранении целостности. Это соответствует требованиям, предъявляемым к описанию модели оборудования. В случае хранения оперативных (диспетчерских) данных, требования к модели изменяются, соответственно отличны и подходы к реализации [7]. При хранении больших объемов данных, таких как данные обследований технологических объектов, когда структура данных фиксирована, а их объем постоянно растет, целесообразно использование классических нормализованных структур.

В целом опыт использования подобной структуры данных для решения практических задач, в частности, при построении системы планирования капитального ремонта компании, осуществляющей магистральный транспорт газа и эксплуатирующей десятки тысячи единиц оборудования, показал [8], что надежность и производительность вполне достаточны для практического применения, а удобство в настройке создает существенные преимущества перед другими вариантами построения модели хранения данных. Расширяемость модели позволила хранить в единой структуре данных как описание оборудования, так и вспомогательную информацию, необходимую для планирования и организации работ, унифици-

цировать инструменты по управлению нормативно-справочной базой, интерфейсные решения. В заключение отметим, что для обеспечения полной совместимости с рекомендациями стандарта ISA-95 предлагаемая реализация может быть легко (без изменений в структуре данных) дополнена недостающими структурами данных по контрольным поверкам.

Список литературы

1. Энциклопедия PLM. Екатеринбург: Азия. 2008.
2. Решетников И.С., Тупышев А.М. и др. Стандарты интеграции многоуровневых информационных систем // Автоматизация в промышленности. № 9. 2009.
3. Зайцев Н.Г. Критический анализ концепции построения баз данных // Управляющие системы и машины, 1983. № 4.
4. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя. М.: ДМК Пресс. СПб: Питер. 2004.
5. Семенов В.А., Морозов С.В., Порох С.А. Стратегии объектно-реляционного отображения: систематизация и анализ на основе паттернов // Тр. Института системного программирования РАН. 2004. Т.8.
6. Фейерштейн С., Прибыл Б. Oracle PL/SQL для профессионалов. 3-е изд. СПб: Питер. 2003
7. Решетников И.С., Коснырев П.Н., Арсланбеков С.А. Рациональный подход к построению диспетчерской системы газотранспортной компании // Автоматизация в промышленности. 2010. № 2.
8. Решетников И.С. Информационная система поддержки принятия решений в многоуровневой структуре на примере организации капитального ремонта нефтегазовой компании // Вестник Казанского государственного технического университета им. А.Н. Туполева. 2007. № 3.

*Решетников Игорь Станиславович — канд. техн. наук, заместитель начальника службы автоматизации, информатизации, телекоммуникаций и связи,
Коснырев Павел Николаевич — начальник отдела внедрения и развития ПО ИУС П
ООО "Газпром центрремонт".*

Контактные телефоны (916) 671-19-74, (499) 580-4-580 доб. 32-986.

E-mail: I.Reshetnikov@gcr.gazprom.ru P.Kosnyrev@gcr.gazprom.ru

Новый 26-дюймовый многофункциональный WUXGA монитор Hatteland для морских применений

Эксклюзивный дистрибьютор Hatteland в России, компания "Ниеншанц-Автоматика", представляет новинку норвежского производителя судового и промышленного компьютерного оборудования — многофункциональный морской WUXGA монитор — JH 26T11 MMD. Новинка выходит на рынок в рамках широкоэкранный серии мониторов Series 1. Основные преимущества этой серии — абсолютная четкость изображения, а также защита экрана от конденсата и ударов благодаря технологии Optical Bonding Technology. Особенность технологии в том, что пространство между матрицей и защитным стеклом заполняется специальным материалом на основе силикона. Таким образом создается единая оптическая среда и устраняется отражение на границах раздела сред, что позволяет практически исключить блики, повысить четкость и контрастность изображения. Кроме того, отсутствие воздуха между стеклом и матрицей повышает защищенность монитора от климатических и механических воздействий.

Новые мониторы поставляются с системой MultiPower, имеющей два встроенных входа для переменного и постоянного тока. MultiPower обеспечивает совместимость мониторов со всеми судовыми системами

питания, автоматически переключая входной канал, если одно из напряжений пропадает или питание имеет нестабильные характеристики.

Другие полезные функции новых устройств: Picture-Vu-Picture позволяет разделить экран монитора на две равные части с возможностью показа в каждой из них изображений с разных видеовходов; функция Picture-In-Picture отображает в одном из углов экрана в небольшом окне изображение с любого видеисточника.

Новые мониторы имеют разъемы: DVI-I, RGB, IEC и USB. Специальный разъем "HATTELAND® Multi-function Connector" позволяет передавать несколько сигналов, таких как композитный видео, последовательный и др. через единственное кабельное соединение, таким образом оптимизируя подачу входных/выходных сигналов.

Устройства серии MMD соответствуют требованиям стандартов ECDIS (Electronic Chart Display and Information System), электронной системы навигации, принятой международной морской организацией (ИМО).

Hatteland планирует производить мониторы и комплектующие данной модели в течение 5 лет, что позволит использовать оборудование в долгоиграющих проектах.

[Http://www.nnz-ipc.ru](http://www.nnz-ipc.ru)

SCADA TRACE MODE поддерживает протокол NMEA 0183

Компания АдАстра (Россия) включила в SCADA-систему TRACE MODE 6 поддержку навигационного протокола NMEA 0183 через бесплатный драйвер.

NMEA (National Marine Electronics Association) — это текстовый протокол связи с навигационным оборудованием. Стал особенно популярен в связи с распространением GPS приемников, использующих этот стандарт. Протокол поддерживается и некоторыми устройствами ГЛОНАСС.

Поддержка протокола NMEA в SCADA TRACE MODE открывает возможность:

- создавать бортовые и навигационные системы на базе SCADA TRACE MODE;
- подключать устройства глобального позиционирования GPS и ГЛОНАСС к АСУТП и системам телемеханики (например, для точной синхронизации времени);
- подключать автоматические метеостанции к ПК.

Новый драйвер доступен начиная с релиза TRACE MODE 6.08. До выхода релиза 6.08 зарегистрированные пользователи могут получить драйвер, обратившись в службу технической поддержки АдАстры.

[Http://www.adastra.ru](http://www.adastra.ru)