ИНСТИТУТ ФИЗИКИ ВЫСОКИХ ЭНЕРГИЙ

ИФВЭ 92-44
ОМВТ

С.В. Клименко, В.И. Ухов

Новые информационные технологии в физике высоких энергий

Протвино 1992

УДК 519.6.00 М-24

Аннотация

Клименко С.В., Ухов В.И. Новые информационные технологии в физике высоких энергий: Препринт ИФВЭ 92-44. – Протвино, 1992. – 25 с., библиогр.: 100.

Обзор содержит информацию об использовании ряда новых информационных технологий в физике высоких энергий. В обзор вошли: объектное программирование, интегрированные системы, UIMS, визуализация, технология знаний, нейронные сети.

Abstract

Klimenko S.V., Ukhov V.I. New Informative Techniques in High Energy Phisics: IHEP Preprint 92-44. Protvino, 1992. – p. 25. refs.: 100.

In this survey a number of new informative techniques applied to HEP are considered: Object-Oriented Programming, Systems Integration, UIMS, Visualisation, Expert Systems, Neural Networks.

Введение

Современные эксперименты в физике высоких энергий используют сложные высокоавтоматизированные установки. Для создания этих установок, для управления ходом эксперимента и для анализа результатов используются все более сложные компьютерные программы.

Создание программ для физики высоких энергий имеет существенную специфику, обусловленную уникальностью разработки. Для каждого эксперимента большой объем программы приходится создавать заново. Лишь возможно их отдельные элементы составляют "постоянный капитал" [7]. Причем это положение усугубляется острым дефицитом времени и людей, способных выполнить такой большой объем интеллектуальной работы. Поэтому здесь необходимо уделять особое внимание эффективности.

Следует подчеркнуть изначальную неопределенность спецификаций программного обеспечения для физических исследований: оно создается, как правило, даже без технического задания — абсолютно необходимого элемента промышленной разработки. Кроме того, система применяемая для физических исследований, — *принципе* должна допускать простое изменение и совершенствование в процессе эксплуатации гораздо в большей степени, чем промышленные системы.

Существенную помощь в создании, сопровождении и модификации *сложных программ могут оказать новые концепции и технологии*, такие, как объектное программирование, интегрированные системы, UIMS, визуализация, технология знаний, нейронные сети и т.п. Некоторые из них сразу разрабатывались (и продолжают разрабатываться) для создания надежных, легко модифицируемых и удобных программ. Другие были созданы в ранних работах по искусственному интеллекту.

И, наконец, третьи отражают насущную необходимость интеграции программного обеспечения.

В данном обзоре перечислен ряд новых информационных технологий, применяемых в физике высоких энергий. В обзор не вошли символьные вычисления, параллельное программирование, использование транспьютеров и др., которые также оказывают большое влияние на программирование и использование компьютеров в физике высоких энергий.

Обзор состоит из трех разделов. В первом, с целью введения используемой терминологии, мы кратко опишем применение компьютеров для задач обработки данных в ускорительных экспериментах[2]. В нем рассматривается: предварительный отбор событий, on line анализ, off line анализ.

Во втором разделе дается краткое описание новых информационных технологий и их возможностей. Он состоит из подразделов: объектное программирование, интегрированные системы, UIMS, визуализация, технология знаний, нейронные сети.

В третьем разделе внимание сконцентрировано на вопросах применения новых информационных технологий в физике высоких энергий. Мы рассматриваем здесь как имеющиеся реализации и проекты, так и ближайшие перспективы. Сюда входят подразделы: триггеры, обработка данных, управление, контроль и поиск неисправностей, проектирование установок, дружественный интерфейс и интеграция программного обеспечения.

Основной целью обзора является распространение информации, связанной с проникновением новых информационных технологий в физику высоких энергий. Данный обзор не содержит подробного анализа затрагиваемых вопросов. Вместо этого он дает ссылки как на ведущиеся, так и на успешно выполненные работы. Он должен рассматриваться скорее как путеводитель.

Основная идея, которую старались подчеркнуть авторы, может быть сформулирована следующим образом: *использование новых информационных технологий позволяет малому числу квалифицированных специалистов создавать большие и надежные программно-аппаратные комплексы.*

[2]Рассмотрено на примере эксперимента *L3* [34]

1. Обработка данных

Характерными для физики высоких энергий являются проблемы обработки больших объемов информации, поступающих с экспериментальных установок. Мы рассмотрим анализ данных на примере эксперимента *L3* [34]. Используемая в нем последовательность обработки информации типична для современных ускорительных экспериментов в области физики высоких энергий.

Обработка состоит из: предварительного отбора событий, on-line анализа и off-line анализа.

1.1. Предварительный отбор событий

Столкновения банчей внутри детектора *L3* (события) происходят с частотой 50 кГц. Каждое регистрируемое событие имеет размер 100 Кбайт. Однако большинство событий не представляет интереса. Интересующие исследователя события происходят примерно раз в секунду. Для того чтобы отобрать интересующие нас события, используется система многоуровневой фильтрации событий, называемая **триггером**.

Триггер 1-го уровня за 10 мкс решает, принадлежит ли данное событие к классу изучаемых. Если принято положительное решение, то включается система полной регистрации события, на что уходит 400 мкс. Что, в свою очередь, приводит к 4% мертвого времени детектора.

Жесткие скоростные требования, предъявляемые к триггеру 1-го уровня, приводят к тому, что он может быть собран на основе аналоговой электроники или на скоростных процессорах со специальной архитектурой и жесткой программой. Кроме того, из-за нехватки времени в триггере не может быть использована информация с различных регистрирующих систем установки. Поэтому для каждой из этих систем может быть разработан отдельный триггер.

Триггер 2-го уровня обрабатывает все события, принятые триггером 1-го уровня. Триггер 2-го уровня состоит из двух частей — 2A и 2B. Триггер 2A решает задачу триггера 1, но только более полно и более точно. Он кончает работу через 100 мкс после столкновения банчей и работает с частотой около 100 Гц. Существует возможность отказаться от события сразу после уровня 2A.

Уровень 2B представляет из себя быстрые программируемые процессоры со специальным набором инструкций. Информация передается через FASTBUS в моде широковещания (broadcast). Что позволяет принимать и обрабатывать ее нескольким процессорам одновременно. Основная задача

триггера 2B — учесть связь между информацией с различных частей детектора *L3*.

Время ответа триггера 2-го уровня составляет 5 мкс. Он пропускает события с частотой около 5 Гц.

Все триггеры 1- и 2-го уровней, как правило, связаны со специальным мини-компьютером, который устанавливает их параметры и контролирует их работу.

Триггер 3-го уровня обрабатывает события, пропущенные триггером 2-го уровня и решает более сложные задачи. Триггер 3-го уровня построен на основе эмуляторов[3]. Эмулятор может проводить сложную и детальную обработку событий по программам, разработанным для "настоящего" компьютера, решая задачи идентификации частиц. Несколько эмуляторов, обрабатывающих события по очереди, с надлежащей буферизацией информации образуют триггер 3-го уровня, который пропускает в среднем одно событие в секунду.

Предварительный отбор событий сокращает объем регистрируемой информации в $10^4 \div 10^6$ раз и позволяет исследовать сравнительно редкие процессы.

После триггера 3-го уровня вся информация о событии, с одной стороны, записывается на магнитные ленты для последующей более полной (и, возможно, неоднократной) обработки, с другой — передается на основной on-line компьютер, где используется для всесторонней обработки хотя бы части принимаемой информации в реальном масштабе времени. Последующая (отложенная) обработка называется off-line обработкой, обработка в реальном масштабе времени — on-line обработкой.

1.2. On-line анализ данных

Цель и смысл on-line обработки — всеобъемлющий контроль установки и хода эксперимента в целом, причем наиболее эффективный вид контроля — контроль по конечным результатам [9].

Используемый метод контроля избавляет от трудоемких исследований зависимостей свойств установки от нескольких тысяч параметров, от необходимости выбирать и устанавливать допуски на изменение этих параметров и комбинации этих изменений. Повышается и надежность установки, так как имеется возможность оперативно принимать реше-

[3]Электронное устройство, эмулирующее команды "большого" широко используемого компьютера

ния о необходимости и целесообразности ремонта при отказе отдельных элементов или о продолжении работы с ухудшенными характеристиками.

Задачи гистограммирования и контроля решаются на машине VAX 11/780 в режиме разделения времени. Пользователю может быть выдана информация о корреляции различных компонент детектора. Для тестирования, калибровки и контроля отдельных компонент детектора используются менее мощные компьютеры, например, VAX 11/750, на которых может быть использовано тоже программное обеспечение, что и на основном компьютере. Все компьютеры объединены в локальную сеть. В результате через терминал основного компьютера можно контролировать ход всего эксперимента.

Однако окончательно все задачи анализа данных нельзя[4] решить в процессе их получения из-за необходимости проведения исследования данных, которое осуществляется итеративно.

1.3. Off–line анализ

Off-line обработка состоит из двух этапов [9]. На первом этапе решаются задачи геометрической реконструкции событий. Второй этап – это статистический анализ.

За год в эксперименте *L3* записывается несколько тысяч лент, что составляет $4 \cdot 10^6$ событий. В подготовку программного обеспечения и анализ данных, получаемых в ходе эксперимента, вовлечены около 100 физиков из 35 институтов в 12 странах. Вычислительная мощность, необходимая для решения этих задач, составляет 12–15 машин типа IBM 370/168. Отдельная проблема, впрочем, в значительной степени уже решенная, состоит в организации надежной связи между коллаборантами.

Что касается области статистической обработки и графического представления данных, то в последнее время здесь стремительно возросла популярность системы PAW [68]. В результате эта система стала de facto стандартом в физике высоких энергий.

2. Новые технологии

В этом разделе дается краткое описание новых информационных технологий и их возможностей.

[4] Имеются эксперименты [23,26], в которых окончательные результаты получаются в конце набора статистики, однако они требуют выполнения очень большого объема работ в процессе подготовки.

2.1. Объектное программирование

Основную роль в появлении объектного программирования сыграли два фактора. *Во-первых*, возросло понимание того, что данные есть не просто нагромождение битов и чисел, а сложные информационные объекты. Эти объекты могут иметь сложную структуру (иерархию), тем не менее часто с ними нужно обращаться как с объектами. *Во-вторых*, было замечено, что над объектами можно выполнять только определенные операции. Признано разумным разрешить доступ к объектам только этим операциям и скрыть их от остальных.[5]

На основе этого были разработаны принципы работы с информацией, имеющей сложную структуру. Следование этим принципам в рамках языка получило название *объектного программирования*. Для обозначения объектного программирования используется и более длинный термин *объектно-ориентированное программирование*. Были созданы и соответствующие языки: Smalltalk, Objective-C, C++, Flavors, ... Многие из них являются расширениями существовавших ранее языков программирования [16,30].

Объектное программирование *значительно увеличивает надежность программ, облегчает отладку и модификацию программ, существенно ускоряет программирование* [29].

К недостаткам объектного программирования следует отнести повышенное потребление ресурсов. Однако и здесь для больших проектов мы можем получить выигрыш за счет большей ясности.

Объектное программирование позволяет небольшому числу профессиональных программистов создавать надежные библиотеки объектов и операций над ними, из которых пользователь мог бы как из конструктора собирать свои программы. Можно надеяться, что это понравится нашим физикам, вынужденным программировать.

Объектное программирование может применяться практически для любых задач и ожидается его широкое применение в физике высоких энергий.

Дополнительную информацию по объектному программированию можно найти в [16,29,30,40,51,52,53].

[5] Аналогичный принцип доступа к информации был применен в методе портов, см. например, [7]

2.2. Интегрированные системы

Подавляющее большинство работ выполняется при помощи нескольких программ. Эти программы должны обмениваться если и не управляющими инструкциями, то, по крайней мере, данными. Кроме того, для работы с ними желательно иметь одинаковый интерфейс. Для решения этих проблем и создаются интегрированные системы. Возможны несколько конструктивных решений.

Во-первых, иногда представляется возможным провести интеграцию в рамках одной программы. Во-вторых, программы могут обмениваться через межпрограммный канал или общую память [6]. Некоторые операционные системы, например VAX/VMS, предоставляют эту возможность. Такой канал также может быть изготовлен самостоятельно [7]. В-третьих, можно воспользоваться "обычным" способом обмена через файлы. В-четвертых, программы могут работать через *систему управления базами данных* (СУБД). Наконец, интересные принципы интегрирования программного обеспечения предложены в Lisp- и Fort системах. Любой из перечисленных выше способов имеет свои преимущества и недостатки. Так, объединение в рамках одной программы через общую память или через межпрограммный канал позволяет создавать более быстрые системы, но существенно затрудняет интеграцию готовых программ, которые вы не можете или не хотите перекомпилировать. С другой стороны, объединение через файлы или СУБД дает больше возможностей для интеграции готовых программ, но система в целом может оказаться слишком медленной из-за большого объема операций ввода-вывода и частого кодирования и декодирования информации. Поэтому часто строят интегрированные системы на основе нескольких подходов.

Дополнительную информацию по вопросам интеграции программного обеспечения можно найти в [4,6,7,16].

2.3. UIMS

Увеличение мощности компьютеров обеспечило адекватный рост затрат на диалоговую компоненту программного обеспечения. Вопрос эффективности использования машин обострился во время стремительного выхода на рынок рабочих станций, объединивших интерактивность с графикой. Термин "эффективность" с тех пор начал менять свое значение: если раньше он отражал такие характеристики, как процессорное время и объем занимаемой памяти, то теперь под ним все чаще понимают простоту разработки, легкость сопровождения и удобство работы с программой.

В настоящее время большие усилия прикладываются к разработке методов и созданию инструментальных средств в рамках систем, получивших название User Interface Management Systems (UIMS).

UIMS — это **инструмент** для разработки программного обеспечения, который управляет всеми коммуникациями между пользователем и прикладной программой. При этом прикладная программа не должна связываться с конечным пользователем напрямую. Цель UIMS — освободить прикладного программиста от технических деталей организации интерфейса, таких, как поддержка окон и графики и учет особенностей устройства.

UIMS позволяет отделить прикладное программирование от разработки пользовательского интерфейса.

UIMS обеспечивает возможность использования различных интерфейсов в одних и тех же приложениях. Выбор интерфейса может зависеть от вкусов и навыков пользователя.

UIMS позволяет использовать один и тот же интерфейс для различных программ и приложений. Такая возможность очень важна для создания интегрированной системы с единым интерфейсом.

UIMS поддерживает *разработку, отладку и исследование* диалога с пользователем.

Как правило, средства UIMS разрабатываются для многооконного операционного окружения, например, такого, как X-Window или OSF/Motif.

Дополнительную информацию по UIMS можно найти в [20,28,33,43,45, 55].

2.4. Визуализация

На проблемы [32], связанные с графическим представлением информации в научных задачах, вновь обратили внимание, когда столкнулись с использованием суперкомпьютеров: сложилась ситуация, когда "исследователь может вычислить больше, чем запомнить, а запомнить — больше, чем понять" (Upson)[32]. Поэтому актуальным становится разработка и использование таких средств, которые бы непосредственно вели от вычислений к пониманию.

Таким средством может быть визуализация (visualisation), понимаемая как способность вызывать зрительные образы [32] в отличие от преобразования чисел в геометрические характеристики (rendering, presentation).

Визуализация призвана "делать видимым невидимое" и является ключевым моментом образного общения [47], опирающегося на два основных принципа: образное мышление и графический дизайн[6]

[6] "графический дизайн — это средство для выражения мысли в визуальной форме

Повышение интереса к визуализации породило в США инициативу ViSC — Visualization in Scientific Computing — визуализация в научных вычислениях [18]. ViSC-инициатива охватывает *следующие сферы* научных исследований: машинную графику, обработку изображений, компьютерное зрение, САПР, обработку сигналов, исследование пользовательского интерфейса. Приложения ViSC также весьма разнообразны. Это — исследования в физике высоких энергий и астрофизике, космические исследования и геологические науки, молекулярное моделирование, визуализация в медицине, исследование структуры и функций мозга, математика и пр.

В настоящее время хорошо развиты методы визуализации 2- и 3-*мерных данных. Представление многомерных данных* гораздо более сложное. Исследования, проведенные Черновым и его коллегами, к сожалению, не дали ощутимого эффекта [5]. Отметим, что при переходе к представлению многомерных данных, даже очевидные вещи не работают. Например, 3-мерное восприятие объекта уже затруднительно из-за принципиальных физиопсихологических ограничений человеческого зрения [48].

Дополнительную информацию по визуализации можно найти в [5,18, 21,24,31,32,47,48].

2.5. *Технология знаний*

Исследования в области искусственного интеллекта привели к разработке компьютерных систем, получивших название "экспертных" или "основанных на знаниях". В их основе лежит представление и использование *формализованных знаний из некоторой предметной области.*

К настоящему моменту технология *экспертных систем* (ЭС) получила широкое распространение. Программа, оформленная в виде ЭС позволяет хранить, тиражировать и использовать знания специалистов. ЭС часто позволяет специалисту контролировать и программировать работу такой системы без помощи программиста.

Теперь о недостатках. Современные ЭС хорошо умеют работать с очень специальными знаниями и плохо с более общими. Например, относительно просто "научить" систему настраивать конкретный спектрометр [54] и гораздо *труднее "обучить" ее настройке спектрометров вообще.* Разработка специализированной ЭС, оправданная в случае серийного производства, может оказаться непропорционально дорогой для уникальной

физической установки. К счастью, в последнее время появились работы по представлению и использованию более общих знаний [38,41]. ЭС могут использовать моделирующую программу для проверки возникающих гипотез [36].

Наиболее удачными средствами создания систем, основанных на знаниях, авторы считают KEE-3, Nexpert Object, Gold Works II и OPS-5. Такие средства обычно называют *оболочками*.

Дополнительную информацию можно найти в [12,13,14,15,16].

2.6. Нейронные сети

Заметный прогресс в исследовании биологических нейронных сетей позволил создать их программные и электронные модели. Далее под термином *нейронные сети* (НС) мы будем иметь в виду программные реализации. Обычно используется упрощенная модель нейронной сети, где "нейроны" расположены по слоям и выходы предыдущего слоя связаны только с входами следующего. Обучение системы состоит в изменении силы связей между нейронами. В рамках такой модели легко построить надежные алгоритмы обучения [50].

Системы, основанные на подходе НС, получают все более широкое распространение. Метод НС позволяет относительно легко строить системы для распознавания ситуаций и оптимизации. Причем такая система может быть обучена на примерах и контрпримерах. Делаются попытки применить НС для решения других задач. Использование специального оборудования, получившего название *нейрокомпьютеров*, существенно ускоряет работу такой системы.

В последнее время появились сообщения о разработке коммерческих систем обмена знаниями между НС и ЭС. Например, между НС Nestor Development System и оболочкой Nexpert Object [19].

Дополнительную информацию можно найти в [50,78,93].

3. Применение технологий

Этот раздел посвящен применению новых информационных технологий в физике высоких энергий. Мы рассматриваем здесь как имеющиеся реализации и проекты, так и ближайшие перспективы.

3.1. Триггеры

В подразделе 1.1 мы уже говорили, что важным элементом экспериментальной установки в физике высоких энергий является **триггер** — специальное устройство, предназначенное для многоуровневой фильтрации событий.

К триггеру предъявляются очень высокие требования. С одной стороны, триггер должен пропускать *все* интересующие нас события, так как строительство и эксплуатация установки стоят дорого. С другой стороны, триггер должен пропускать *только* интересующие нас события, так как увеличение потока пропущенных событий на большинстве уровней требует дополнительных затрат и может привести к увеличению мертвого времени установки, т.е. триггер должен работать как можно точнее. Кроме того, триггер должен работать быстро. В итоге постепенный рост сложности экспериментов ведет к усложнению триггеров [35].

В настоящее время для создания триггеров разного уровня предлагается использовать как НС, так и ЭС.

Использование НС позволяет легко создать триггер, обучающийся на примерах [59,78,93]. Такой триггер можно обучить на программе, моделирующей, например, методом *Монте Карло*, работу установки. Другим достоинством НС является возможность использования нейрокомпьютера, значительно уменьшающего время отклика. В процессе работы (например, на основании on-line контроля) можно подстраивать такой триггер, указывая ему на "правильные" и "неправильные" события.

Неспособность НС к самостоятельным логическим рассуждениям и наше неумение строить сложные НС затрудняет их использование для триггеров верхнего уровня. Тем не менее, если удастся провести декомпозицию задачи, то с ней смогут справиться несколько НС. Другим недостатком НС является необходимость переобучения всякий раз, когда нужно изменить триггер. Это может оказаться очень неудобным на стадии наладки.

ЭС также могут быть применены для создания триггеров [63,74,94]. Они очень хорошо умеют работать с формальными знаниями, и это дает возможность использовать их для триггеров верхнего уровня. В отличие от НС, знания в ЭС могут быть представлены в форме близкой человеку, что делает несложной процедуру модификации триггера. Знания, представленные в ЭС, можно развернуть в оптимизированное *дерево решений* и использовать даже для "быстрых" триггеров нижнего уровня [74,94]. Реализация триггера на основе параллельных таблиц решений описана в [2].

ЭС с трудом обучаются на примерах и плохо умеют работать на "параллельном" оборудовании.

Итак, НС и ЭС хорошо умеют справляться с недостатками друг друга. К сожалению, еще остаются определенные трудности с их интеграцией. Работы ведутся по двум направлениям: создание систем передачи знаний между НС и ЭС, организация тесного взаимодействия НС и ЭС.

Системы быстрого управления триггером и информирования пользователя о его работе требуют совершенного интерфейса. Эта проблема детально обсуждается в подразделе 3.5.

3.2. *Обработка данных*

Для облегчения процесса обработки данных создаются интегрированные системы с хорошо организованным интерфейсом [7,58,61,68,69].

Осознание того, что пользователь-физик работает не с файлами и *программами*, а с *некими информационными* объектами, над которыми выполняются вполне определенные действия, позволило создать интерактивные системы очень удобные в обращении, например, систему *"Reason"*[7]

Создание подобных систем стало возможным еще и благодаря увеличению мощности используемых компьютеров и наличию у них графических возможностей, а также достижениям в области визуализации, таким, как способы представления 2- и 3-мерных данных и иконографика.

Некоторые трудности в создании таких систем вызваны возможностью использовать привычное программное обеспечение, такое, как пакеты ZEBRA, GEANT ..., написанное на Фортране и желанием использовать современные объектные языки программирования. Сейчас разрабатываются системы анализа данных декларативного типа, в которых пользователь описывает *не как* делать, *а что* делать [65,98]. Разрабатываются интерактивные языки и системы для поиска треков и реконструкции событий [7,70]. Так как работа по поиску треков и реконструкции событий довольно монотонна, а объемы ее значительны, то здесь особенно необходимы хороший интерфейс и контроль оператора.

Значительную помощь в обработке данных могут оказать ЭС. Во-первых, их можно использовать для проверки целостности данных, поступающих с установки [66]. Во-вторых, ЭС можно использовать для реконструкции сложных событий [82]. В-третьих, ЭС может выполнять

[7]Девизом создателей этой системы было *"Look and Feel"* [61], что можно перевести как *"Воспринимай!"*. В системе использована иконографика и способы представления многомерных гистограмм.

функции управления в сложной, многопроцессорной и многозадачной системе интерактивного отображения и анализа событий [65].

Для реконструкции треков и событий могут быть использованы НС [62,93]. Наличие у них способностей к оптимизации и распознаванию образов позволяет им относительно легко справляться с этой задачей, в то время как для обычных программ и ЭС эта задача часто оказывается трудно разрешимой. Важное преимущество НС, реализованных на нейрокомпьютерах, - это скорость. Так, в [93] рассматривается проект on-line поиска треков и реконструкции всех событий.

3.3. Управление, контроль и поиск неисправностей на больших установках

Физические установки могут быть очень сложными. Поэтому для эффективной работы с ними нужны не менее сложные системы управления. Эксперимент OPAL [88], например, содержит 160000 каналов сбора данных. Система обеспечения безопасности на такой установке, как LEP (ЦЕРН) тоже должна работать с большими объемами данных и знаний [72]. Ну а задача управления современными ускорителями без автоматических систем управления вообще решена быть не может.

В системах столь больших размеров даже при высокой надежности их элементов могут часто возникать неполадки. Для контроля за правильностью работы как самой установки, так и ее информационной части должны быть созданы специальные системы. Системы обнаружения неисправностей могут существенно повысить точность и эффективность работы сложной установки.

Технология ЭС позволяет создать надежные и удобные в работе системы управления, контроля и диагностики. Возможность "открытого" представления знаний разрешает специалистам-физикам самим менять эти системы. Важной частью таких систем является организация диалога с пользователем.

Мы были вынуждены объединить в одном подразделе вопросы управления установками, контроля их работы и поиска неисправностей. Дело в том, что большинство рассматриваемых нами систем опираются на знания, применимые как для управления, так и для диагностики. Заметим также, что если установка перестала правильно реагировать на управляющие воздействия, то, следовательно, она неисправна. Поэтому большинство систем управления являются одновременно и системами контроля, но не наоборот. Так, в [90] описана система, выполняющая только функции контроля.

В настоящее время ЭС могут осуществлять контроль правильности работы установки [36,60,64,66,88,72,81,85,36,90].

Кроме контроля ЭС могут управлять установкой [36,49,54,73,85,88] и решать задачи поиска неисправностей [36,41,60,63,64,75,76,81,91].

ЭС, основанные на традиционном подходе, когда вы работаете со знаниями нижнего уровня, т.е. знаниями о поведении конкретного устройства, могут быть использованы для работы со стандартными [54,75] или, в крайнем случае, с постоянными частями установок [60,72,73,85]. К сожалению, физические установки уникальны и подвержены непрерывным переделкам. Поэтому разработка традиционной ЭС может оказаться для них слишком долгой и дорогой. Для управления однотипными, но отличающимися установками необходимо научить систему переводить знания о конструкции и элементах конкретной установки в алгоритмы управления этой установкой и знания о ее функционировании. Такие ЭС были разработаны [63,76,91]. Этот подход представляется авторам наиболее перспективным.

Технология ЭС дает возможность сохранять знания и навыки управления и работы с физической установкой при смене персонала [73]. Для сложных и долго живущих установок это немаловажный фактор.

При проверке гипотез, выдвигаемых ЭС, могут быть использованы моделирующие программы для уменьшения нагрузки на оборудование [36, 63,76,91]. Моделирующие программы могут использоваться как в режиме off-line для накопления знаний о поведении конкретной установки, так и прямо в режиме on-line. В первом случае в процессе управления не требуется постоянно запускать большую моделирующую программу, но зато нужно работать с большими объемами знаний.

Для работы с большими объемами данных и знаний ЭС может использовать возможности СУБД, например, Oracle [41,72,81]. ЭС могут быть разработаны при помощи специальных средств (оболочек) [41,49].

Рассмотрим в качестве примера работу [41], в которой описан прототип ЭС для диагностики канала инжекции в PS Booster (CERN). Наряду с качественными моделями в данной работе сделана попытка использования согласованных с ними количественных моделей. Такой подход позволяет соединить достоинства традиционной теории управления с возможностями технологии знаний.

В качестве исходной информации для пополнения базы знаний была использована база данных под управлением СУБД

Oracle, хранящая описание канала инжекции. Специальная программа транслировала эту информацию в набор правил и фреймов для ЭС. Таким образом, изменения, вносимые в оборудование (и заносимые в базу данных), могли быть автоматически отражены в базе знаний ЭС.

Использование Influence Net для создания качественной модели установки, несомненно, упростило ЭС и повысило ее "интеллект". Однако, если неисправность в установке вызвана не отклонением параметра, а изменением функции элемента (например, ошибкой в полярности корректора), то обычная Influence Net может перестать соответствовать моделируемой установке, а ее использование может стать невозможным. В такой ситуации более гибкой оказалась бы составная Influence Net, собранная из отдельных сменных моделей для каждого элемента. Тогда, заменив модель неисправного элемента, можно восстановить соответствие и продолжить работу.

Для работы со знаниями было использовано мощное программное средство KEE, что значительно облегчило разработку и отладку ЭС. К сожалению, эксплуатация разработанной ЭС также предполагает использование KEE. Это может сильно замедлить работу ЭС или даже сделать невозможной ее эксплуатацию в режиме реального времени. Ограничение по скорости может помешать сделать ЭС "умнее", так как рост базы знаний ведет к замедлению.

Значительный интерес представляют неоконченная работа по внесению в ЭС знаний об осциллограммах, выдаваемых диагностирующим оборудованием.

Известны примеры решения задач диагностики с помощью средств образного анализа [10,11,17].

НС могут быть привлечены для управления сильно нелинейными устройствами [83]. Использование НС для диагностики представляется не столь эффективным, однако и здесь может сыграть свою роль их быстродействие. Нейронная сеть, натренированная быстро распознавать определенные виды неисправностей, могла бы быть полезна для ускорения работы системы диагностики. Ускорение будет особенно заметным, если большой процент отказов вызван небольшой группой поломок.

3.4. Проектирование установок

Для проектирования и расчетов будущей установки широко используются как общие средства автоматизации проектирования, так и специальные моделирующие программы. Ожидается разработка интегрированных систем [25], в которых системообразующим элементом могла бы быть среда конструктивной геометрии.

Предлагается использовать подход ЭС для согласования проекта при внесении в него изменений [44], а также для автоматического проектирования подсистем установки [46]. Проблема автоматической генерации проекта в общем случае является достаточно сложной, однако возможно построение систем, которые бы собирали проектируемую установку из готовых элементов. В качестве примера здесь можно привести известную систему R1, составляющую конфигурации вычислительных машин серии VAX в зависимости от требований заказчика [22] или ЭС для проектирования и моделирования сбора данных [79].

Существенную помощь в разработке и модификации программного обеспечения могут оказать интеллектуальные средства поддержки сложных программных проектов [86].

Работы по проектированию установки и программного обеспечения ведутся интерактивно. Поэтому следует особенно подчеркнуть важность удобного и дружественного интерфейса. Эта проблема включает в себя не только удобные интерфейсы отдельных программ, но также унификацию интерфейса в интегрированной системе. Использование достижений визуализации при разработке интерфейса может значительно поднять производительность и надежность разрабатываемой системы. Для качественной визуализации проектируемых установок или их частей могут быть разработаны специальные пакеты [27].

3.5. Дружественный интерфейс и интеграция программного обеспечения

В последнее время было, наконец, обращено внимание на разработку удобных и эффективных интерфейсов пользователя и на объединение программного обеспечения в физике высоких энергий [58,60,61,63,65,67,68,69, 70,80,84,87,88,89,96,97,98,99].

Интеграция программного обеспечения и разработка удобного интерфейса в сущности решают одну задачу — сделать работу с компьютером более быстрой, более удобной и более надежной.

Важным вопросом объединения программ является выбор языка (языков) программирования. С одной стороны, хочется выбрать Фортран, так как именно на Фортране доступны различные физические пакеты (ZEBRA, GEANT, MINUIT ...) и программное обеспечение установок, созданных ранее [69,80,84]. С другой - новые объектные языки позволяют достичь большей надежности программ и значительно ускоряют их разработку, отладку и модификацию [61,65,79,88,96,97]. Объектное программирование вынуждает создавать независимые программные модули, которые могут быть использованы впоследствии.

В ЦЕРНе значительные объемы информации об устройстве ускорителя и установок, а также текущие значения параметров многих систем хранятся под управлением СУБД Oracle. Благодаря этому идет процесс постепенной интеграции программного обеспечения через СУБД [72,81,92].

ЭС могут организовывать взаимодействие нескольких программ и пользователя [65,89]. Другим их применением могло бы быть оказание помощи в подготовке сложных заданий, например, в выборе распадов для программ моделирования.

Напомним, что появились системы декларативного типа, где пользователь описывает *не как* делать, *а что* делать [65,98]. Удобные интерфейсы были разработаны для интегрированных систем [58,61,68]. Особенно хотелось бы отметить использование технологии UIMS в работе [58] и последовательного объектного подхода в работе [61].

Необходимо обратить внимание на использование достижений в области визуализации для разработки удобных интерфейсов. Такие работы, как исследование данных в off-line анализе или проектирование установок осуществляются *интерактивно* и *итеративно*. Производительность таких работ может быть повышена применением иконографики, например, для обозначения доступных операций над данными [61,42] или для обозначения доступных элементов конструкции. Удачный или неудачный выбор образов для представления многомерной информации, такой, как экспериментальные данные или характеристики альтернативных проектов, может повлиять не только на скорость принимаемых решений, но и на их качество.

Заключение

Некоторые задачи, например, создание высокоэффективной системы сбора информации в экспериментах на SSC и LHC [35], без использования новых информационных технологий будут заведомо неразрешимы.

Скорость решения других, таких, как анализ данных, диагностика аппаратуры и т.д. благодаря этим технологиям может быть значительно увеличена.

Отдельно следует напомнить об опасности возникновения большого технологического разрыва между мировым научным сообществом и отечественной экспериментальной физикой высоких энергий. Такой разрыв может сильно затруднить научное общение и привести к исключению нас из мировой научной кооперации.

Для создания сложных высокоавтоматизированных экспериментальных установок, для управления ходом эксперимента и для анализа результатов используют все более и более сложные компьютерные программы.

Создание программ для физики высоких энергий имеет существенную специфику, обусловленную уникальностью разработки и ограничением в сроках. Кроме того, системы применяемые для физических исследований, *в принципе* должны допускать простое изменение и совершенствование в процессе эксплуатации гораздо в большей степени, чем промышленные системы. Поэтому здесь необходимо уделять особое внимание эффективности программирования.

Мы надеемся, что новые информационные технологии найдут себе применение в отечественной физике высоких энергий. Мы также надеемся, что данный обзор будет этому способствовать.

Напомним, что основной целью авторов было дать ссылки на работы, проводимые в этой области.

Список литературы

[1] Антипов Ю.М. и др. - Препринт ИФВЭ 76-129, Серпухов, 1976.

[2] Балдин Б.Ю. и др. *Устройство быстрого триггера на базе параллельных таблиц решений для работы в условиях повышенной множественности регистрируемых событий*: Препринт ИФВЭ 90-146, Протвино, 1990.

[3] Баурн С. *Операционная система UNIX*. - М., Мир, 1986.

[4] Броуди Л. *Начальный курс программирования на языке ФОРТ*. - М., Финансы и статистика, 1990.

[5] Гришин В.Г. *Образный анализ экспериментальных данных*. - М., Наука, 1982, 237 с.

[6] Дейтел Г. *Введение в операционные системы.* – М., Мир, 1987, 2 т.

[7] Клименко С.В. *Программные системы обработки информации с больших физических установок ИФВЭ и их применение для анализа экспериментальных данных*: Диссертация на соискание ученой степени доктора физико-математических наук. – Серпухов, ИФВЭ, 1985.

[8] Клименко С.В. и др. – Препринт ИФВЭ 76–164, Серпухов, 1976.

[9] Лебедев А.А. *Экспериментальные исследования образования J/ψ – частиц в адрон–ядерных взаимодействиях на ускорителе в ИФВЭ*: Диссертация на соискание ученой степени доктора физико-математических наук.– Серпухов, ИФВЭ, 1980.

[10] Ремизов В.В., Сула А.С., Михайлов В.В. *Применение систем образной диагностики в массовой эксплуатации авиационных силовых установок.*– В сб.: Тезисы докл. конф. ОБРАЗ–90 (Суздаль).– М., 1990.

[11] Строганов М.П., Берестень М.Т. *Человекомашинный метод анализа результатов диагностического эксперимента.* – В сб.: Тезисы докладов конф. ОБРАЗ–90 (Суздаль).– М., 1990.

[12] Уотерман Д. *Руководство по экспертным системам.* – М., Мир, 1989.

[13] *Представление и использование знаний.* /Под ред. Х. Уэно, М. Исидзука.– М., Мир, 1989.

[14] *Приобретение знаний* /Под ред. Х. Уэно, Ю. Саэки.– М., Мир, 1990.

[15] *Построение экспертных систем* /Под ред. Ф. Хейес-Рот, Д. Уотерман, Д. Ленат. – М., Мир, 1987.

[16] Хювенен Э., Сеппянен И. *Мир Лиспа.* – М., Мир, 1990, 2 т.

[17] Шерстюк И.Н. *Образы оперативного мнемо-языка для АСУТП энергоблоков.*– В сб.: Тезисы докл. конф. ОБРАЗ–90 (Суздаль).– М., 1990.

[18] *ACM SIGGRAPH.* Computer Graphics. 1987, Vol. 21, N6.

[19] AI Expert, Dec., 1989.

[20] Allari S. et al. *Achievements Derived from the Adoption of UIMS with Graphic Interaction Techniques in Vitamin Project.* Proceeding of the Graphics and Interaction in ESPRIT Session, Eurographics'89.

[21] Arnold D.B. *CGI Versus X-11. State of the Art Reports.* Eurographics'89.

[22] Bachant J., McDermott J. *R1 revisited: four years in the trenches.* The AI Magazine, vol. 5, no. 3, Fall 1984.

[23] Barrelet E. et al. - Preprint CERN EP 80 93. Geneva, 1980.

[24] Bergeron R.D., Grinstein G.G. *A Reference Model for the Visualization of Multi-Dimensional Data.* - EG'89.

[25] Brun R., Carminati F., Chernyaev E., Perevozthikov V. *A new geometry description for GEANT.* Draft material.

[26] Campbell M.K. et al. - Preprint BNL-29789, Upton, 1981.

[27] Chernyaev E.V., Obraztsov V.F., Petrovykh Yu.L., Samarin A.V. *Detector Visualization Package.* DELPHI note 87-30, PROG 74. 1987.

[28] Cockton G. *Interaction Ergonomics, Control and Separation: Open Problems in User Interface Management Systems.* Prep., AMU8811/03H, Scottish HCI Centre, February 1988.

[29] Cox B.J. *Object-Oriented Programming, An Evolutionary Approach.* Addison-Wesley Pub. Co., 1986.

[30] Eckel B. *Using C++.* Osborne/McGraw-Hill, 1989.

[31] Hewitt W.T. *PHIGS BR. State of the Art Reports.* EG'89, 1989. - 64 p.

[32] Hubbold R.J. *Interactive Visualization: Challenges and Prospects.* State of the Art Reports. - EG'89, 1989. - 39 p.

[33] Kilgour A. *Theory and practice in user interface management systems:* Information and Software Technology, vol. 29, no. 4, May 1987.

[34] *L3. Technical Proposal.* CERN, May 1983.

[35] Lankford A.J. *Computing and Data Handling Requirements for SSC and LHC Experiments.* SLAC-PUB-5243, May 1990.

[36] Lee M. *Model-based Expert System for Linac computer controls.* Prep. SLAC-PUB-4724, September 1988.

[37] Lee M., Clearwater S., Kleban S., Selig L. Error-finding and error-correcting methods for the start-up of the SLC. Prep. SLAC-PUB-4214, February 1987.

[38] Lenat D. *EURISKO: A Program That Learns New Heuristics and Domain Concept.* Artificial Intelligence 21 (1983) 61-98.

[39] Lenat D. *Why AM and EURISKO Appear to Work.* Artificial Intelligence 23 (1984) 269-294.

[40] Lieberherr K., Holland M. *Assuring Good Style for Object-Oriented Programs.* IEEE Software, Sept. 1989.

[41] Malandain E. *Application of a diagnostic Expert System in the Accelerator domain.* Prep. CERN/PS 89-52 (OP), September 1989.

[42] Marcus A. *Designing Graphical User Interfaces.* Unix World, August 1990.

[43] Myers B. *Creating dynamics interaction techniques by demonstration.* - ACM CHI 87-GI Conference, 1987.

[44] Prevereaud J. *Systemes experts de Cao: ils sont prets!.* Bureaux d'etudes Automatismes, no. 33, 1987.

[45] Prime M. *User Interface Management Systems — A Current product Review.* - Computer Graphics Forum 9, 1990.

[46] Rabaey J. *Silicon Compilation and Design Synthesis for Digital Systems.* Proceeding of 1988 CERN School of Computing.- CERN 89-06, Geneva 1989.

[47] Schmid C., Schmid C. *Handbook of Graphic Presentation.* - John Willey, 1979.- 308 p.

[48] Scientific American, August 1988.

[49] Schultz D., Brown P. *The Development of an Expert System to Tune a Beam Line.* Nuclear Instruments and Methods in Physics Research A293 (1990) 486-490, Elsevier Science Publishers B.V.

[50] Simpson P. *Neural Networks : Research and Applications Series.* New York, Pergamon Press,1990.

[51] Software Test Lab : *An objective look at C++ environments.* IEEE Software, March 1989.

[52] Software Test Lab : *Three object-oriented environments for the desktop.* IEEE Software, May 1989.

[53] Wolf W. *A Practical Comparison of Two Object-Oriented Languages.* IEEE Software, Sept. 1989.

[54] Wong C., Crawford R., Kunz J., Kehler T. *Application of artificial intelligence to triple quadrupole mass spectrometry.* Proceeding of the IEEE Nuclear Science Symposium, San Francisco CA, October, 1983.

[55] Ziegler J. *Direct Manipulation Techniques for Human-Computer Interfaces.* - Eurographics'90. Technical Report Series.

[56] *New Computing Techniques in Physics Research.* DU CNRS, Paris, 1990. *Proceeding of first International Workshop on Software Engineering, Artificial Intelligence and Expert System for High Energy Physics. Lyon (France), March 19-24, 1990.*

[57] Aarnio P., Hakulinen T. *Developing a personal computer based expert system for radionuclide identification* [Lyon-90].

[58] Alberty J., Ans 'mo F., Barberio E., O. Di Rosa, B. van Eijk, Commare G., Marino M., Cifarelli L., Matsuura T., Meng R. *An integration of physics packages data modelling and human interface* [Lyon-90].

[59] Altherr T. *Neural Networks and Calorimeter Cluster Recognition* [Lyon-90].

[60] D'Antone I., Mandrioli J., Matteuzzi P. *Expert system strategies for the diagnostic in a particle physics experiment* [Lyon-90].

[61] Atwood B., Blankenbecker R., Kunz P., Mours B., Weir A. *The Reason Project* [Lyon-90].

[62] Awes T. *A Neural Networks Approach to the Problem of Photon Pair Combinatorics* [Latin-90].

[63] Bizzarri L., Corazziari F., Falciano S., Gualaccini P., Luminari L., Savarese M., Trasatti E. *Control and Diagnosis of the L3 en[illegible] trigger* [Lyon-90].

[64] Block F., Becks K., Hemker A. *FB Expert : an expert system for management and error diagnosis of FUSTBUS networks* [Lyon-90].

[65] Bonissent A., Delfino M., Etienne F., Qian Z. *Loosely coupled distributed architecture for Interactive event Display and Analysis* [Lyon-90].

[66] Bonissent A., Mathieu O., Delfino M. *An Expert System for off-line data validation* [Lyon-90].

[67] Brettler B. *[illegible] Objective-C* [Lyon-90].

[68] Brun R., Couet O., Vandoni C., Zanarini P. *Visualization of Scientific Data for High Energy Physics : PAW , a Gernal-Purpose Portable Software Tool for Data Analysis and Presentation* [Lyon-90].

[69] Bruyan F. *Design of the L3 off-line Software Base* [Lyon-90].

[70] Burnett T. *IDAL : An interactive analysis language for high energy physics* [Lyon-90].

[71] Caillian R. *Languages for HEP* [Lyon-90].

[72] Chevrier F. *Safety Controlled by an Expert System on Experimental Sites in High Energy Physics* [Lyon-90].

[73] Clearwater S., Cleland W., Provost F., Stern E., Zhang Z. *A Real-Time Expert System for Experimental High Energy,'Nuclear Physics* [Lyon-90].

[74] Clearwater S., Cleland W., Stern E. *Use of Learning programs for SSC Trigger Strategy Studies* [Lyon-90].

[75] Corazziari F., Falciano S., Luminari L., Savarese M., Trasatti E. *An Intelligent Tool for Fault Diagnosis in FASTBUS systems* [Lyon-90].

[76] Dague P., Deves P., Luciani P., Taillibert P. *Model-Based Diagnosis of Analog Systems : an Application to Electronic Circuits* [Lyon-90].

[77] Dauboin P. *A Knowledge Based system for Nuclear Plant Loading pattern* [Lyon-90].

[78] Denby B. *Neural Network Tutorial for High Energy Physicist* [Lyon-90].

[79] Durr E. *Expert system for designing data acquisition systems based on the VME bus* [Lyon-90].

[80] Fisher S. *LIPS : a logical analyser and personal simulator for use in a Forth environment* [Lyon-90].

[81] Le Goff J. *Use of an Expert System to Monitor and Control a Large Hep. Energy Physics Experiment at CERN. The L3 Slow Control* [Lyon-90].

[82] Hemker H. *Linking of symbolic and subsymbolic mechanism* [Lyon-90].

[83] Howell J. *Neural Networks in the Ion scans Control Environment* [Lyon-90].

[84] Knoblock J. *Methods of Software Development for the ALEPH Experiment* [Lyon-90].

[85] Lai C. *An on line gas control system using an artificial intelligence language PROLOG II* [Lyon-90].

[86] Lamb D., Jain N. *An Overview of Knowledge Based Tools for Software Development* [Lyon-90].

[87] Meyer B. *Eiffel : An Introduction* [Lyon-90].

[88] Middleton R. *Control of a LEP experiment using a knowledge based system* [Lyon-90].

[89] Moreira de Souza J., Cavalcanti da Rocha A. *TABA HEP : a heuristic workstation for developing HEP software* [Lyon-90].

[90] Olson D., Greiman W., Hall D., Baldauf D., Day C. *Experience using an automated fault location system with a time of flight wall detector array* [Lyon-90].

[91] Ortmann Y., Becks K., Brand K., Hemker A. *DELPHI EXPERT: an expert system for error diagnosis in high energy physics detectors* [Lyon-90].

[92] Palermo L., de Souza J. *Recording DELPHY Off-line Production Activity in a Meta Database* [Lyon-90].

[93] Peterson C. *Neural Networks and High Energy Physics* [Lyon-90].

[94] Proriol J., Jousset J., Guicheney C., Falvard A., Henrard P., Pallin D., Perret P., Prulhiere F. *Event recognition : multivariate analysis methods to tag b quarks events in ALEPH* [Lyon-90].

[95] Remaund B., Calvez J. *ADA as a tool for simulation on massive parallel systems and for hard real-time application* [Lyon-90].

[96] Roussean B. *Building the Scientist Computing Environment.* [Lyon-90].

[97] Sibomana M., Longree Y., Marechal P., Nemry M., Ninane A., Somers F. *Direct manipulation user interfaces* [Lyon-90].

[98] Werner C., Pimenta M., Varela J., Souza J. *Fado tagging system : decoupling logic from code* [Lyon-90].

[EPAC-90] *EPAC-90. 2nd European Particle Accelerator Conference.* Proceeding, Edition Frontiers, 1990.

[99] Lutz J., Marsaudon J., Kapps E., Persigny J. *The Vivitron Process Control* [EPAC-90].

[100] Strubin P., Fietier N. *Using Distributed Intelligence to Control The Vacuum System of a Very Large Accelerator* [EPAC-90].

Рукопись поступила 16.03.92

С.В. Клименко и др.
Новые информационные технологии в физике высоких энергий..

Оригинал-макет подготовлен с помощью системы LaTeX
Редактор М.Л. Фоломешкина. *Технический редактор Л.П.Тимкина.*

Подписано к печати 24.03.1992 г. Формат 60 × 90/16.
Офсетная печать. Печ.л. 1.56. Уч.-изд.л. 1.65. Тираж 260. Заказ 191.
Индекс 3649. Цена 2 руб. 50 коп.

Институт физики высоких энергий, 142284, Протвино Московской обл.

2 руб. 50 коп. Индекс 3649

П Р Е П Р И Н Т 92-44, И Ф В Э, 1992