

## АДАПТИВНОЕ УПРАВЛЕНИЕ СТРУКТУРОЙ ГАРАНТОСПОСОБНЫХ УЧЕТНЫХ СИСТЕМ

*В работе описана концепция построения динамической структуры учетных информационных систем, использование которой позволяет повысить показатели их гарантоспособности. Концепция заключается в хранении программного кода основных алгоритмов обработки данных в составе информационной базы системы*

*Ключевые слова: учетная информационная система, гарантоспособность, динамическая структура.*

Учетные информационные системы (УИС) получили в настоящее время повсеместное распространение. Этому способствуют, с одной стороны, большое количество предприятий в Украине, а с другой - законодательно закрепленная необходимость ведения каждым предприятием бухгалтерского и налогового учета.

Одним из основных требований, которое предъявляется к УИС есть необходимость постоянного поддержания уровня ее гарантоспособности. Под гарантоспособностью понимается свойство системы оказывать предусмотренные услуги, которым можно оправдано доверять. Гарантоспособность включает такие качества системы, как безотказность, готовность и обслуживаемость в период эксплуатации [1]. Качество безотказности характеризуется вероятностью безотказной работы, готовность – вероятностью застать УИС в работоспособном состоянии в данный момент времени, а обслуживаемость – временем, которое требуется для выполнения изменений в системе при изменении законодательства [2]. Последнее качество для УИС является определяющим, так как постоянное внесение изменений в законодательную базу или в предметную область (ПрО) обуславливает необходимость постоянных изменений учетных систем во время их эксплуатации. Алгоритмы работы УИС, зависящие от законодательства будем называть критическими алгоритмами. Необходимость изменения критических алгоритмов снижает гарантоспособность системы, приводит к повышению расходов общественно-необходимого времени на сопровождение УИС, угрожает большими финансовыми потерями и приостановкой производственных процессов одновременно на большом количестве предприятий. Приведенные положения позволяют

отнести учетные информационные системы к классу критических систем.

Адаптирование УИС к изменениям ПрО во время сопровождения осуществляется сегодня путем передачи пользователям обновленных критических алгоритмов в виде файлов, которые имеющих объем порядка 10 – 15 мегабайт. При этом возникает чисто техническая проблема, связанная с тем, что необходимость одновременной передачи большого объема информации большому количеству пользователей входит в конфликт с ограниченной пропускной способностью каналов связи [3]. Это делает невозможным своевременное получение пользователями УИС требуемых обновлений. Для решения проблемы адаптируемости требуется значительное (на несколько порядков) сокращение объема необходимой для обновлений информации.

Большие объемы передаваемых файлов обусловлены тем, что основные алгоритмы УИС, которые обеспечивают функциональность и жестко связаны с действующим законодательством (ПрО), хранятся в системе в виде откомпилированных исполняемых кодов (одной динамической библиотеки или одного исполняемого файла). Поэтому в рамках существующих управляющих структур УИС решение этой проблемы не представляется возможным.

Существующие традиционные подходы к построению управляющих структур УИС являются статичными, то есть не позволяют осуществлять замену отдельных критических алгоритмов и не удовлетворяют требованиям обеспечения гарантоспособности систем. Такая статичная структура построения существующих УИС представлена на рис. 1.

Все существующие УИС построены таким образом, что основные алгоритмы в них реализованы традиционным путем – как составная часть общего скомпилированного программного кода системы. В базе данных содержатся только поля, в которых хранятся числовые значения параметров этих алгоритмов, что позволяет выполнить только частичную их настройку на изменение ПрО. В случае таких изменений каждый отдельный критический алгоритм не подлежит замене у пользователя – заменить можно только весь программный блок, в состав которого входит этот алгоритм. В силу этого такие традиционные структуры можно считать статичными.

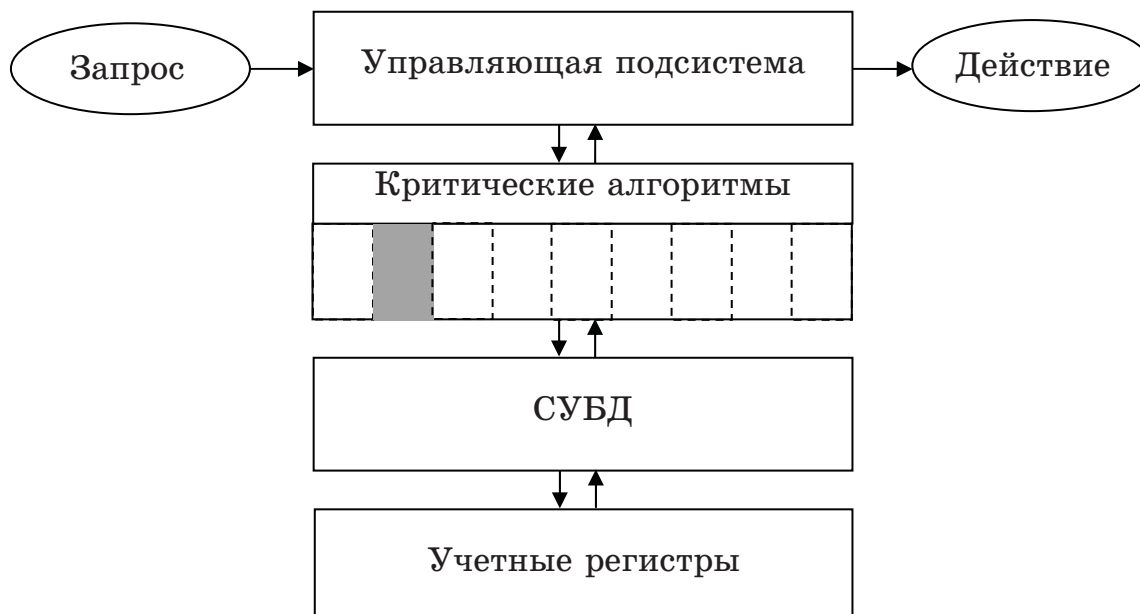


Рисунок 1 - Традиционная статическая структура построения ОИС

Построение УИС на базе традиционной статичной структуры имеет два существенных недостатка. Во-первых, как уже было отмечено, статичная структура ухудшает показатель обслуживаемости. Во-вторых, ухудшаются, также, показатели безотказности и готовности за счет больших потерь времени и низкой надежности при передаче пользователю файлов большого. Как следствие этого, снижается и гарантоспособность УИС.

Решение проблемы повышения гарантоспособности возможно за счет предложенного в работе нового подхода к построению динамической структуры УИС. Такая динамическая структура УИС представлена на рис. 2.

Эта структура позволяет динамически изменять любые отдельные критические алгоритмы, не изменяя при этом скомпилированный программный код системы. Основной особенностью предложенной структуры является хранение критических алгоритмов не в общем программном коде, а в информационной базе системы в виде не транслированного или частично транслированного кода (так называемого «р-кода»). Такой код занимает гораздо меньше места, чем соответствующий ему оттранслированный двоичной код. Это решает проблему его быстрой передачи к большому количеству пользователей и значительно уменьшает время простоев системы при ее адаптации. Благодаря этому улучшается показатель обслуживаемости системы и как следствие – ее гарантоспособность.

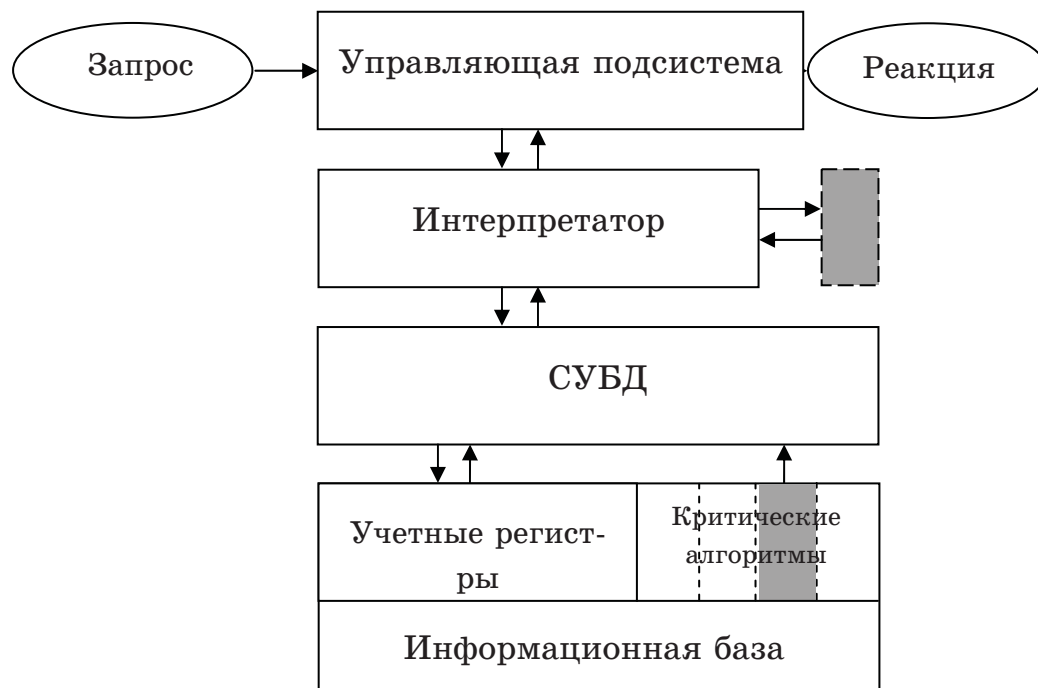


Рисунок 2 - Динамическая структура построения гарантоспособных УИС

Работа УИС с динамической структурой происходит следующим образом. Система получает запрос пользователя на выполнение действия, которое требует работы критического алгоритма. Этот запрос анализируется управляющей подсистемой, которая определяет, какой именно критический алгоритм должен быть использован для обработки этого запроса. Управляющая подсистема выдает соответствующую информацию в СУБД, которая находит в информационной базе нужный программный код и передает его на выполнение интерпретирующей подсистеме. Интерпретатор выполняет этот программный код. Как результат его выполнения может изменяться состояние учетных регистров системы (ее информационная база) или формируются отчеты для пользователя.

Использование предложенного подхода и построение динамической структуры УИС имеет такие преимущества:

1. Программный код критических алгоритмов хранится в информационной базе системы. При изменении условий Про замена критического алгоритма сводится к простому изменению текстовых полей информационной базы. Такие изменения легко могут быть реализованы в автоматическом режиме. Основной программный код УИС при этом остается неизменным.

2. За счет того, что в информационный базе хранятся неоттранслированные коды алгоритмов, их размер представляет несколько де-

сятков килобайт, что делает возможной его передачу к большому количеству пользователей в небольшой промежуток времени и не требует значительного трафика во время передачи данных от разработчика к пользователям. Это приводит к сокращению времени на обслуживание системы и повышает ее гарантируемость.

Недостатком предложенного подхода может считаться необходимость применения интерпретатора и связанное с этим возможное снижение быстродействия системы. Однако проведенные исследования показали, что такое снижение быстродействия не является существенным.

По данным литературных источников можно сделать вывод о том, что современные интерпретирующие системы, такие как С# и Java по основным показателям быстродействия несущественно уступают компиляторам (Visual C, Intel C). По опубликованным данным тестирования (таблица 1), средние показатели быстродействия указанных интерпретаторов лишь на 25% уступают соответствующим компилирующим системам, а для такого широко используемого компилятора, как Borland Delphi эти показатели вообще на 7% меньше, чем у интерпретаторов [4].

Принимая во внимание то, что быстродействие не является решающим показателем учетных систем, такое его уменьшение можно считать полностью приемлемым. А с учетом того, что наиболее распространенная на сегодня платформа для создания учетных систем – 1С:Предприятие – построена как интерпретирующая, то для нее сравнение быстродействий систем, построенных на основе старой и новой структур управления вообще не имеет смысла.

Описанная динамическая структура управления реализована в созданной под руководством авторов УИС «АгроКомплекс», предназначенной для автоматизации всех видов учета на предприятиях сельского хозяйства. Опыт внедрения этой УИС на десяти сельскохозяйственных предприятиях Украины и практика трехлетней эксплуатации показала эффективность предложенного подхода.

## Сравнение быстродействия компиляторов и интерпретаторов

Язык программирования	Тесты и время их выполнения в секундах								Среднее значение
	Вызов метода класса	Вызов вирт. метода класса	Доступ к членам класса	Быстрая сорт.	Пузырьковая сорт.	Tree sort	Обработка данных типа string	Обработка данных типа float	
<b>Компиляторы</b>	<b>6.20</b>	<b>6.67</b>	<b>4.81</b>	<b>10.50</b>	<b>7.39</b>	<b>11.85</b>	<b>6.81</b>	<b>8.26</b>	<b>7.81</b>
Visual C 6.0	5.78	6.49	5.06	8.7	5.02	11.85	6.08	12.26	7.65
Intel C++	6.32	6.50	5.04	9.20	4.85	12.31	3.39	0.29	5.98
Borland Delphi	6.50	7.01	4.34	13.59	12.31	11.40	10.97	12.24	9.80
<b>Интерпретаторы</b>	<b>3.18</b>	<b>10.85</b>	<b>4.19</b>	<b>11.00</b>	<b>7.83</b>	<b>19.90</b>	<b>3.43</b>	<b>12.61</b>	<b>9.12</b>
C#	2.16	7.22	3.54	9.50	5.29	23.60	3.38	12.24	8.38
Java	4.20	14.48	4.85	12.50	10.37	16.20	3.48	12.98	8.88

## ЛИТЕРАТУРА

1. Харченко В.С. Гарантоспособность и гарантоспособные системы: элементы методологии //Радиоелектронні і комп'ютерні системи.– 2006.– Вип.5(17). – С.7-19.
2. Скляр В.В. Оценка качества и экспертиза программного обеспечения. Лекционный материал. / В.В. Скляр; под ред. Харченко В.С.– Министерство образования и науки Украины, Национальный аэрокосмический университет им. Жуковского Н.Е. «ХАИ», 2008, – 204с.
3. Урывский Л.А. Зависимость пропускной способности дискретного канала связи от его энергетического потенциала при использовании многопозиционных сигналов. / Л.А. Урывский, Е.А. Прокопенко // Науково-виробничий збірник «Наукові записки Українського науково-дослідного інституту зв'язку». №1(9), 2009, С. 41 - 50
4. Чистяков В. Современные средства разработки: сравнение производительности. / В. Чистяков. // Технология «Клиент-сервер», № 3, 2001, С. 18-48
5. Маєвський Д.А. Інформаційна система «Аг-роКомплекс» для бухгалтерського та оперативного обліку у сільському господарстві / Д.А. Маєвський, Т.Я. Тінтулова, В.М. Антощук. // Сб. Аграрний вісник причорномор'я. Технічні науки. 2009, - Вип. 48, С. 151 – 156