

## МИНИМИЗАЦИЯ МЕЖМОДУЛЬНОГО ИНТЕРФЕЙСА ДЛЯ ОБЕСПЕЧЕНИЯ НАДЕЖНОСТИ МУЛЬТИВЕРСИОННОГО ПРОГРАММНОГО КОМПЛЕКСА\*

И. В. Ковалев, Ю. А. Нургалева, А. В. Шахматов, С. А. Чекмарев, Ф. А. Лукин

Сибирский государственный аэрокосмический университет имени академика М. Ф. Решетнева  
Россия, 660014, Красноярск, просп. им. газ. «Красноярский рабочий», 31. E-mail: kleniks@yandex.ru

*Показано, что использование модульного принципа на этапе технического проектирования мультиверсионного программного комплекса связано с процессом оптимизации состава и взаимосвязей его отдельных компонентов. Предложена оригинальная постановка задачи синтеза мультиверсионной модульной структуры программного комплекса, обеспечивающей минимизацию межмодульного интерфейса. Представленная постановка задачи позволяет сформировать мультиверсионный набор компонентов модулей, использующихся в программах комплексах как произвольно, так и в заданной заранее последовательности. При этом набор мультиверсий удовлетворяет заданным условиям, обеспечивая максимальную надежность ПК и минимальное значение характеристик межмодульного интерфейса.*

*Ключевые слова:* мультиверсионный программный комплекс, модульная структура, межмодульный интерфейс, надежность.

## MINIMIZATION OF INTER-MODULE INTERFACE FOR PROVISION OF RELIABILITY OF MULTIVERSION SOFTWARE SYSTEM

I. V. Kovalev, J. A. Nurgaleeva, A. V. Shahmatov, S. A. Chekmaryov, F. A. Lukin

Siberian State Aerospace University named after academician M. F. Reshetnev  
31 "Krasnoyarskiy Rabochiy" prosp., Krasnoyarsk, 660014, Russia. E-mail: kleniks@yandex.ru

*It is shown that the use of a modular principle at the stage of technical design of multiversioning software package is connected to the process of optimization of structure and interactions of the individual components. The authors offer an original definition of the problem of synthesis of multiversioning modular structure of software system that ensures the minimization of inter-module interface. The presented definition of the problem allows to generate a set of components multiversioning modules that are used in the programs of the complexes as arbitrary, and in the pre-set sequence. In this case the set of multiverses meets the specified conditions, ensuring maximum reliability and minimum PC specifications of inter-module interface.*

*Keywords:* multiversioning software system, modular structure, intermodule interface, reliability.

При синтезе мультиверсионной архитектуры программного обеспечения (ПО) важным этапом является распределение моделей и методов оценки параметров надежности по фазам жизненного цикла разработки программных средств систем управления и обработки информации [1]. Наиболее важными являются фазы, которые вносят наибольший вклад в надежность ПО: фаза дизайна архитектуры, фаза кодирования, фаза тестирования компонентов и фаза тестирования системы.

Под архитектурой ПО, как и вообще любых других средств обработки информации, в узком смысле понимают совокупность их свойств и характеристик, призванных удовлетворить запросы пользователей. В более широком смысле архитектура ПО – обобщенная модель информационной системы, достаточная для понимания принципов ее функционирования [2].

Надежность архитектуры включает в себя надежность индивидуальных элементов. Сбой отдельного

элемента приводит к неработоспособности этого и, возможно, других элементов, но не всей системы в целом. Сбой в системе или ее функциях может возникнуть в периоде простоя системы. Период простоя системы определяется как отрезок времени, на котором система или ее часть не может выполнять функции. Период простоя системы приводит к резкому снижению производительности, поэтому уменьшение времени простоя системы является одним из наиболее важных факторов при разработке архитектур распределенного программного обеспечения.

Основные понятия теории надежности комплексов программ базируются на понятиях теории надежности, первоначально развившейся применительно к аппаратным комплексам, однако имеются существенные различия в принципах обеспечения надежности программного обеспечения и других технических систем.

\*Работа выполнена в рамках ФЦП «Научные и научно-педагогические кадры России», ГК № 14.B37.21.0451.

Эксперименты показывают, что применение избыточности на самых ранних этапах разработки, насколько это возможно, уменьшает вероятность появления ошибок. Надежное ПО можно создать с помощью тщательной разработки архитектуры и выявления ошибок в компонентах, которые оказывают самое большое влияние на надежность системы. Эти компоненты определяются как наиболее часто используемые или архитектурно связанные со множеством других компонентов.

Модульная организация предполагает деление ПО на функционально завершенные части (модули), унификацию связей между ними и установление иерархии взаимодействия компонентов, которая определяется последовательностью их вызова. Вызов осуществляется передачей управления от вызывающего модуля к вызываемому, который по окончании выполнения возвращает управление вызывающему модулю. Вызов группы программ осуществляется передачей управления диспетчеру группы программ более низкого уровня иерархии, входящей в данный программный комплекс (ПК).

Основные методы современной практической разработки программных комплексов базируются на функциональной декомпозиции с использованием модульно-иерархических принципов [3]. При этом на каждом иерархическом уровне ПК ограничивается сложность компонентов и их связей. В результате общая сложность программной системы с возрастанием объема задач растет значительно медленнее, чем при неструктурированном проектировании.

Очевидно, что использование модульного принципа на этапе технического проектирования связано с процессом оптимизации состава и взаимосвязей отдельных компонентов мультиверсионного программного и информационного обеспечения систем обработки информации и управления. Таким образом, комплекс задач синтеза модульных систем на основе мультиверсионной программной архитектуры включает в себя оптимальный выбор состава модулей мультиверсионного ПК и информационных массивов, содержания межмодульного интерфейса, а также структуры системы в целом.

Известно, что основными критериями синтеза модульных систем обработки информации и управления на этапе технического проектирования являются минимум сложности межмодульного интерфейса, минимум времени обмена между оперативной и внешней памятью ЭВМ при решении задач, минимум объема неиспользуемых данных при пересылках между оперативной и внешней памятью ЭВМ; минимум технологической сложности алгоритмов обработки данных, являющейся обобщением показателя транспортно-фактора при реализации алгоритмов решения функциональных задач; максимум информационной производительности модульных систем при решении задач, максимум достоверности обработки данных [4].

В качестве ограничений при решении задач синтеза оптимальных модульных систем мультиверсионной обработки информации и управления, кроме перечисленных показателей, используются такие характеристики, как состав процедур и объем каждого модуля, состав и объем информационных массивов,

сложность и состав интерфейса между отдельными модулями, степень дублирования процедур и информационных элементов, объем оперативной памяти ЭВМ, число каналов связи с внешними запоминающими устройствами, возможности передачи управления внутри и между модулями, последовательность выполнения процедур, допустимые затраты и время разработки и внедрения системы и т. д.

Исходными данными для решения задач синтеза являются техническое задание на разработку системы, результаты анализа существующей системы управления и результаты ее предварительного обследования. На основе этих данных определяются следующие характеристики: множество функциональных задач обработки данных; множество процедур обработки данных, включая альтернативные (реализация мультиверсионности); множество информационных элементов системы, связанных с процедурами обработки данных и идентифицируемых по типам: входной, промежуточный, выходной; варианты возможного взаимодействия процедур обработки данных с информационными элементами; технологическая матрица смежности либо технологическая скелетная матрица; множество допустимых последовательностей выполнения процедур; характеристики процедур, информационных элементов и технических средств [4].

Решение поставленных задач позволяет разработать функциональную блок-схему мультиверсионной модульной структуры ПК на архитектурном (системном) уровне, обеспечивающую минимальный интерфейс между модулями комплекса. Следует отметить, что дублирование мультиверсионных элементов (мультиверсий) в модулях проектируемого ПК обеспечивает снижение значения показателя программно-информационной связи этих модулей.

Рассмотрим постановку задачи синтеза мультиверсионной модульной структуры ПК, обеспечивающей минимизацию межмодульного интерфейса [5].

В проектируемый модуль добавляются мультиверсии (не меньше двух), и известны следующие значения:

$R_{ij}$  – надежность  $j$ -й версии  $i$ -го модуля;

$X_{ij}$  – булева переменная, равная 1, если  $j$ -я версия используется в  $i$ -м модуле, или 0 – в противном случае;

$n$  – количество модулей;

$m_i$  – количество версий  $i$ -го модуля;

$R_i$  – надежность  $i$ -го модуля:

$$R_i = 1 - \prod_{j=1}^{m_i} (1 - R_{ij})^{X_{ij}}, \quad i = \overline{1, n}.$$

Синтезируются программы и указываются последовательности, определяющие, какие модули и в каком порядке используются, при этом:

$K$  – количество программ;

$S_k$  – множество модулей, соответствующих  $k$ -й программе.

Затем, задав ограничения на надежность системы и ограничения на межмодульный интерфейс, введем дополнительные переменные:

$F_k$  – частоту использования  $k$ -й программы;

$R$  – надежность системы, максимизация которой осуществляется согласно выражению

$$\max R = \sum_{k=1}^K F_k \prod_{i \in S_k} R_i.$$

Межмодульный интерфейс – это количество одних и тех же мультиверсий модулей, использующихся разными программами [6].

Для формализации постановки задачи, обеспечивающей минимизацию межмодульного интерфейса, введем дополнительную переменную  $W_{ik}$  – булеву переменную, равную 1, если  $i$ -й модуль используется  $k$ -й программой и 0 – в противном случае.

$$Y_{kj} = \begin{cases} 1, & \text{если } \sum_{i=1}^n W_{ik} X_{ij} \geq 1, \\ 0, & \text{если } \sum_{i=1}^n W_{ik} X_{ij} = 0. \end{cases}$$

Переменная  $Y_{kj}$  служит для формализации взаимосвязи разрабатываемой системы с отдельными мультиверсиями модулей.

Для минимизации межмодульного интерфейса на уровне отдельных версий в качестве критерия используется следующее выражение:

$$\min \sum_{j=1}^J \sum_{k=1}^{K-1} \sum_{k'=k+1}^K Y_{kj} Y_{k'j}.$$

Представленная постановка задачи позволяет сформировать мультиверсионный набор компонентов модулей, использующихся в программах модульного ПК как произвольно, так и в заданной заранее последовательности. Полученный набор мультиверсий будет удовлетворять заданным условиям, обеспечивая максимальную надежность ПК и минимальное значение характеристик межмодульного интерфейса.

#### Библиографические ссылки

1. Ковалев И. В., Капулин Д. В., Царев Р. Ю. Архитектурная надежность программного обеспечения

информационно-управляющих систем : монография / Краснояр. гос. аграр. ун-т. Красноярск, 2011.

2. Kovalev I. V. System of Multi-Version Development of Spacecrafts Control Software. Sinzheim, 2001.

3. Kovalev I. V., Younoussov R. V. Fault-Tolerant Software Architecture Creation Model Based on Reliability Evaluation // Advances in Modeling and Analysis. D. 2002. Vol. 7, № 3. P. 31–44.

4. Мамиконов А. Г., Кульба В. В. Синтез оптимальных модульных СОД. М. : Наука, 1986.

5. Капчинский И. А., Кузнецов А. С., Штенцель А. В. Принципы формирования мультиверсионного программного комплекса // Вестник СибГАУ. 2008. Вып. 1 (18). С. 18–23.

6. Новой А. В., Штенцель А. В. Оценка надежности мультиверсионной программной архитектуры систем управления и обработки информации // Вестник СибГАУ. 2008. Вып. 3 (20). С. 73–79.

#### References

1. Kovalev I. V. Kapulin D. V., Carev R. J. *Arhitekturnaja nadezhnost' programmnogo obespechenija informacionno-upravljajushih system* (Architectural software reliability management information systems). Krasnojar. gos. agrar. un-t. Krasnojarsk, 2011, 182 p.

2. Kovalev I. V. System of multi-version development of spacecrafts control software. Pro Universitate Verlag Sinzheim. 2001, 77 p.

3. Kovalev I. V., Younoussov R. V. Fault-tolerant software architecture creation model based on reliability evaluation. Advances in Modeling and Analysis – D. 2002, vol. 7, № 3, pp. 31–44.

4. Mamikonov A. G., Kulba V. V. *Sintez optimal'nyh modul'nyh SOD* (Synthesis of optimal modular SOD). Moscow, Nauka, 1986.

5. Kapchinskij I. A., Kuznecov A. V., Shtendel A. V. *Vestnik SibGAU*. 2008, № 1 (18), pp. 18–23.

6. Novoj A. V., Shtendel A. V. *Vestnik SibGAU*. 2008, № 3 (20), 73–79.

© Ковалев И. В., Нургалева Ю. А., Шахматов А. В., Чекмарев С. А., Лукин Ф. А., 2013

УДК 519.87

## О ПАРАМЕТРИЧЕСКОМ МОДЕЛИРОВАНИИ СТОХАСТИЧЕСКИХ ОБЪЕКТОВ

А. А. Корнеева, Е. А. Чжан

Сибирский федеральный университет  
Россия, 660041, Красноярск, просп. Свободный, 79. E-mail: ekach@list.ru

*Рассматривается проблема идентификации многомерных статических объектов при различном уровне априорной информации. Исследуется вопрос о влиянии выбранной параметрической структуры модели на качество идентификации в условиях помех, ее соответствии реальному математическому описанию объекта. Приводятся результаты численного исследования различных параметрических моделей многомерных статических процессов.*

*Ключевые слова:* априорная информация, идентификация в узком и широком смысле, стохастический объект, непараметрические и параметрические модели.