В. Сафонов, И. Сафонов
(Международный институт науки единства,
Арлингтон, Виржиния, США)

РИСКИ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ОПТИМИЗАЦИЯ ПОТОКОВ РАБОТ

Поведенческий подход позволил нам системно объединить, а часто и синергетически соорганизовать, многие традиционные методологии обеспечения безопасности и оптимизации функционирования структурно-алгоритмических систем со случайными нарушениями [38-43]. Этот симбиоз ***упреждающего Инжиниринга Доверия*** и ***реагирующего Риск Менеджмента*** создаёт условия для наиболее эффективного обеспечения не только безопасности, но и других важных свойств (надёжности, точности, секретности, производительности и т. п.) в общих рамках жизненного цикла проектирования и совершенствования систем. При этом реабилитируется прикладная математика, частично дискредитированная негибким применением (однокритериального!) математического программирования и в ещё большей степени расслабляющим влиянием диктата (к сожалению, не всегда инженерно образованных) разработчиков и программистов. Известно, что решения, основанные на математике, более долговечны, чем решения, основанные на технологии. Достаточно вспомнить булеву алгебру, теорию графов, формулу Байеса и дискретные аналоги дифференциального исчисления. Для образованного и опытного инженера простота и точность прикладной математики очевидны. Наша концепция делает проектные решения обоснованно независимыми от влияния поставщиков и строго ориентированными на интересы заказчика.

Моделирование и оптимизация помогают с достаточной точностью понять проблемы и наилучшим образом решить их. В этом случае неадекватное использование терминов "моделирование" и "оптимизация" является не только нежелательным, но и недопустимым. К сожалению, любое иллюстративное и неполное описание (эскиз, набросок) проблемы, как правило, называется "моделью" проблемы, а любой шаг кажущемся правильным в направлении улучшения обычно называется "оптимизация". Сомневающиеся в состоятельности моего утверждения могут легко его проверить путём поиска в Интернете соответствующих терминов и анализа их интерпретаций. Сказанное выше в первую очередь относится к тысячам рекламных сообщений о методах и инструментах управления потоками работ оптимизации деловых процессов. Один из авторов является пионером оптимизации потоков работ, мы рады возможностям соответствующего многомиллиардного рынка [http://www.internettime.com], но как профессионалы мы огорчены дискредитацией терминов и концепций моделирования и оптимизации в этом контексте. Производительность и качество, надёжность и долговечность, безопасность и секретность ответственных

процессов и технологий нуждаются в доверии к их создателям и пользователям, равно как и к советникам и консультантам руководителей. Убедитесь сначала, что вам предлагают действительно ***лучшие (строго обоснованные) решения.*** Мы сможем гарантировать нашим клиентам корректное моделирование и оптимальное решение их проблем, если таковые в принципе разрешимы и когда эти решения приемлемы для клиентов даже с учётом причин, иногда выходящих за рамки технического задания. Проблемы же однокритериальной (всё реже!) и многокритериальной (всё чаще!) оптимизации, основанной на персонально-ориентированном моделировании ситуаций и процессов для автоматизации бизнеса и обработки данных, решаются в четырёхмерном пространстве ***целей, ресурсов, структуры*** и ***поведения,*** первично определяемых и часто корректируемых клиентами. Поскольку однокритериальные проблемы (с нетривиальным множеством ограничений) до настоящего времени являются массовыми, а также в силу строгости и простоты их формулировок самими заказчиками, начнём именно с них. Многокритериальные проблемы оптимизации потоков работ будут нами рассмотрены в последующих публикациях.

Двумя основными жизненно важными функциями человека в его взаимодействии с природой, технологиями и себе подобными являются ***принятие решений,*** акты и процессы которого будем обозначать маленькими буквами латинского алфавита $\{a_1, a_2,\ldots, b_1, b_2,\ldots, c_1,\ldots\}$, и ***выполнение действий,*** акты и процессы которого будем обозначать прописными буквами латинского алфавита $\{A_1, A_2,\ldots, B_1, B_2,\ldots, C_1,\ldots\}$. Элементарными (атомарными) актами принятия решений и выполнения действий являются ***решения*** (a) и ***действия*** (A), соответственно, рассматриваемые как неделимые на рационально установленном для каждой задачи уровне абстракции её моделирования, анализа, оптимизации и синтеза. Синтаксис, семантика и прагматика корректной композиции решений и (или) действий определяют с необходимой степенью детализации дискретные и непрерывно-дискретные процессы человеческой активности, как правило, более или менее автоматизированной. Если эти процессы описывают (моделируют) материальные, энергетические, финансовые, информационные и (или) иные производства и (или) услуги, то соответствующие решения и действия принято называть ***работами,*** а сами процессы моделирования принятия решений и (или) выполнения действий – ***потоками работ***. Анализ возможных ситуаций, оптимизация и синтез потоков работ являются стадиями ***проектирования потоков работ,*** равно как контроль, оптимизация и коррекция выполнения отдельных работ или их взаимосвязанных совокупностей – стадиями ***управления потоками работ***.

Проектирование доверия и ***управление риском*** являются частными, но важными случаями интегрированного процесса проектирования потоков работ и управления ими, а поскольку риск и неопределённость являют-

ся неизбежными атрибутами сложных потоков работ, то и важность ***оптимального проектирования доверия и управления риском*** не вызывает сомнения. Чем сложнее поток работ, тем менее наглядным становится его описание и более сложной – его модель. Чем выше уровень автоматизации потока работ, тем меньше смысла имеет визуализация этого потока для его автоматизированного (а иногда и автоматического) проектирования и управления. Принцип Эшби ("Теория сложных систем есть теория упрощения") постепенно вытесняется принципом Глушкова ("Система управления должна быть по крайней мере той же сложности, что и управляемый объект"). Для инструментов автоматизированного проектирования и управления определяющим их эффективность критерием становится уровень формализации объектов и процессов проектирования и управления, а их наглядность уходит на второй план, вытесняясь требованиями повышенных строгости и точности моделирования.

И, наконец, для оптимизации необходимы соответствующие метрики с известными или вычислимыми зависимостями между ними. Поэтому формальные (алгоритмические, алгебраические) модели потоков работ предпочтительнее графических, что конечно не исключает и последние. Подобного же рода соображения и накопленный опыт говорят в пользу теоретико-игровых моделей и методов по сравнению с аналитическими, и многокритериальных и "размытых" – по сравнению с классическими моделями и методами математического программирования, при анализе и оптимизации процессов принятия решений.

Предположим, что имеется несколько (j) альтернатив реализации каждого функционального (***внешнего***) или аспектного (***внутреннего***) действия $A(i)$ потока работ (алгоритма, процедуры) A, представляющих также альтернативные реализации всего требующего оптимизации потока работ $A(i)$ -> { $A(i, j)$ }, $j = 1, 2, \dots, n(i)$, $j = 1, 2, \dots, N$. Каждая возможная реализация действия $A(i, j)$ характеризуется вектором параметров (характеристик, критериев) производительности, качества, надёжности, безопасности и т. п. { $f(i, j)$, $r(i_1, j)$, $r(i_2, j)$, ... , $r(iS, j)$ }, где f – функциональные (***целевые***) и r – аспектные (***обеспечивающие свойства***) требования или ограничения; и весь поток работ A оценивается вектором параметров { F, R_1, R_2, ... , R_S }, где $F = F(f_1, f_2, \dots, f_N)$, $R_k = R_k(r_{1k}, r_{2k}, \dots, r_{Nk})$. Необходимо выбрать наилучшую ***каноническую*** (последовательную, параллельную, альтернативную, и (или) циклическую) ***композицию*** (архитектуру) реализационных альтернатив действий $A(i)$, $i = 1, 2, \dots, N$, потока работ A, которая в результате приводит (это не самый общий, но достаточно часто используемый на практике, случай) к глобальному экстремуму целевой функции (функционала) F при заданных ограничениях вектора параметров этого потока работ:

F -> extr, $R_k <= R_k0$, где R_k0, $k = 1, 2,\dots, S$, заданы.

Наиболее часто используемыми параметрами являются Время Выполнения Потока Работ (Performance), Готовность и (или) Вероятность Правильного Выполнения (Reliability), Уровни Безопасности (Safety) и Секретности (Security), Сложность (Complexity), Цена (Price), Эффективность Капиталовложений (Return on Investment – ROI), etc. Очевидно, что каждый из этих параметров (а также количество типов функциональных и аспектных операторов – при необходимости их унификации) может быть выбран в качестве целевой функции F. В этом случае остальные параметры должны удовлетворять установленным ограничениям. Известно, что любой поток работ может быть представлен композицией названных выше четырёх канонических форм – последовательной (линейной), параллельной (конъюнктивной), альтернативной (дизъюнктивной) и (или) циклической (итеративной).

Как правило, виды функциональных зависимостей $F = F(f_1, f_2, \ldots, f_N)$ и $R_k = R_k(r_{1k}, r_{2k}, \ldots, r_{Nk})$ для канонических форм уже известны или тривиально определимы, а для произвольного потока работ определяются его алгоритмической структурой. В самом общем случае, для решения оптимизационных задач может быть применён метод неявного перебора, использующий схему ветвей и границ (подробнее в [17]). В большинстве практически важных случаев оптимизация потоков работ может быть упрощена путём учёта специфики конкретных деловых, производственных, технологических, информационных и иных процессов. Например, градиентные [13] или аналитические [1, 2] методы находят широкое применение при проектировании в условиях дефицита производительности (Design for Performance), безопасности (Design for Safety), секретности (Design for Security) и (или) надёжности (Design for Reliability). В более сложных ситуациях мы использовали выпуклое программирование [10], метод ветвей и границ [15], динамическое программирование [11] и другие методы. Применимость моделей и методов оптимизации проверялась путём специальных исследований их адекватности конкретным ситуациям [10] и устойчивости к неточности исходных данных [2]. Мы предлагаем унифицированную (единую) методологию и формализованные методы для оптимального удовлетворения требований к безопасности, секретности и надёжности алгоритмов деловых процессов и программ атоматизации этих процессов. В основу этой методологии положены концепция и архитектура структурно-алгоритмической оптимизации [25]. При этом учитываются принципиальные отличия между проектными и операционными (эксплуатационными) аспектами безопасности, секретности и надёжности, а также учитывается корреляция между этими аспектами. Это приводит к необходимости разделения интересов (а также полномочий), связанных с влиянием неадекватности моделирования, проектных и программных ошибок, несовершенства защиты и злонамеренных вторжений, неисправностей и сбо-

ев оборудования и недооценкой опасности действий собственного персонала и масштабов возможных материальных потерь.

Проблемы аспектной оптимизации безопасности, защищённости и надёжности неизбежно фокусируются на сбалансированности с требованиями и ограничениями к параметрам производительности, точности, (потенциальных) возможностей, энергопотребления, стоимости (цены) и других. Формализованы и исследованы проблемы устойчивости используемых моделей, методов и программ к информационным флюктуациям, ситуациям принятия решений и процедурам выполнения действий. Предлагаются актуальные постановки и эффективные методы тестирования (не только автоматизированных) деловых процессов и процессов автоматизированной обработки данных с корректирующей избыточностью и без неё. Оригинальное программное обеспечение оптимизации потоков работ находится в процессе разработки. Наша методология находит широкое коммерческое применение при разработке автоматизированных систем технологического и организационного управления, САПР, инструментов прогнозирования процессов и поддержки решений, специализированных управляющих компьютеров, средств обработки документов, мультипроцессорных систем, поисковых машин и т.п.

Литература

1. Бондарь Ю. и Сафонов И. В. Метод оптимального использования алгоритмической избыточности. – Управляющие системы и машины, 1975, № 3.

2. Бондарь Ю. и Сафонов И. В. Устойчивость модели оптимального обнаружения и исправления ошибок. – Разработка и применение АСУ и средств автоматизации. – Киев: Институт автоматики, 1977.

3. Борисюк А. А., Гук К. Н., Коссовский В. Г. и Сафонов И. В. Обеспечение надёжности СЦВМ управляющего комплекса. – Диагностика, контроль, надёжность систем управления. – Киев: Институт автоматики, 1976.

4. Буч Г. Объектно-ориентированное программирование с примерами применения. – Москва: Конкорд, 1992. – 519 с.

5. Глушков В. М. Теория автоматов и микропрограммные алгебры. – Кибернетика, 1965, № 1.

6. Глушков В. М. Теория автоматов и формальные преобразования микропрограмм. – Кибернетика, 1965, № 6.

7. Глушков В., Барабанов А., Калиниченко Л., Михновский С. и Рабинович З. Вычислительные машины с развитыми системами интерпретации. – Киев: Наукова Думка, 1970.

8. Глушков В. М., Капитонова Ю. В. и Летичевский А. А. Автоматизация проектирования вычислительных машин. – Киев: Наукова Думка, 1975.

9. Глушков В. М., Цейтлин Г. Е. и Ющенко Е. Л. Алгебра. Языки. Программирование. – Киев: Наукова Думка, 1974.

10. Демьянчук А. П. И Сафонов И. В. Применимость метода выпуклого программирования для надёжностной оптимизации систем реального времени. – Диагностика, тестирование и надёжность управляющих систем. – Киев: Институт автоматики, 1976.

11. Демьянчук А. П., Карась В. М. и Сафонов И. В. Надёжностная оптимизация

алгоритмов АСУ методом динамического программирования. – Проблемы надёжности систем и средств автоматики. – Киев: Техника, 1976.

12. Капитонова Ю. В. и Летичевский А. А. Математическая теория проектирования вычислительных систем. – Москва: Наука, 1988.

13. Карась В. М. и Сафонов И. В. Обеспечение надёжности функционирования и развития АСУ (Оптимизация алгоритмов). – Киев: РДНТП, 1974.

14. Коссовский В. Г. и Сафонов И. В. Формализованное надёжностное проектирование специализированных цифровых машин. – Проблема надёжности систем управления. – Киев: Наукова Думка, 1973.

15. Куриленко В. Т., Левченко С. Н. и Сафонов И. В. Надёжностная оптимизация алгоритмов методом ветвей и границ. – Алгоритмы, программное и техническое обеспечение автоматизированного управления производством. – Киев: Институт автоматики, 1978.

16. Макаренко Г. И. и Сафонов И. В. Система производства программ для управляющих ЦВМ. – Логическое управление. Вып. 3. – Москва: Энергоиздат, 1981.

17. Мамиконова О. А. и Сафонов И. В. Оптимизация структурированных алгоритмов и программ. – Средства реализации систем программирования. – Киев: ИК АН УССР, 1983.

18. Мурса Л. Г. и Сафонов И. В. "Селена": кто она такая? – Искусство кино, 1988,
№ 11.

19. Редько В. Н. Дескриптологические основания программирования. – Кибернетика и системный анализ, 2002, № 1.

20. Сафонов И. В. К вопросу введения корректирующей избыточности путём преобразования алгоритмической структуры цифрового автомата. – Теория автоматов, Вып. 1. – Киев: ИК АН УССР, 1969.

21. Сафонов И. В. Об одном способе задания исходной информации при формализованном надёжностном синтезе. – Четвёртая республиканская научная конференция молодых исследователей по системотехнике. II том. – Киев: ИК АН УССР, 1969.

22. Сафонов И. В. О формализованном надёжностном синтезе дискретных устройств. – Кибернетика, № 5, 1971.

23. Сафонов И. В. Об алгоритмическом этапе формализованного надёжностного проектирования ЦВМ. – Управляющие системы и машины, № 1, 1972.

24. Сафонов И. В. Оптимизация при автоматизированном проектировании систем управления. – Автоматизированное проектирование АСУ. – Москва: Финансы и статистика, 1981.

25. Сафонов И.В. Надёжностное проектирование алгоритмов управления. – Владивосток: Институт автоматизации и процессов управления, 1982.

26. Соложенцев Е. Д., Карасев В. В. и Соложенцев В. Е. Логико-вероятностные модели риска в банках, бизнесе и качестве. – Санкт-Петербург: Наука, 1999.

27. Backus J. Can programming be liberated from the Neumann style? A functional style and its algebra of programs. – Comm. ACM, 1978, 21(8). – P. 613-641.

28. Bollella, et al. The Real-Time Specification for Java. – Boston, MA: Addison-Wesley, 2000.

29. Booch, Grady, James Rumbaugh, and Ivar Jacobson. The Unified Modeling Language User Guide. – Reading, MA: Addison-Wesley, 1999.

30. Dijkstra, E. W. On the Role of Scientific Thought. – E. W. Dijkstra Archive (EWD447), August 1974.

31. Fowler, Martin, et al. Refactoring: Improving the Design of Existing Code. – Boston, MA: Addison-Wesley, 2000.

32. Gamma, Erich, et al. Design Patterns: Elements of Reusable Object-Oriented Software. – Reading, MA: Addison-Wesley, 1995.

33. Henderson-Sellers, Brian. Object-Oriented Metrics: Measures of Complexity. – Upper Saddle River, NJ: Prentice-Hall, 1996.

34. Hoch, Detlev J., et al. Secrets of Software Success. – Boston, MA: Harvard Business School Press, 1999.

35. Kiczales, G., J. Lamping, A. Mendhekar, C. Maeda, C. Lopes, J.-M. Loingtier, and J. Irwin. Aspect-Oriented Programming. In ECOOP'97 – Object-Oriented Programming, 11th European Conference, LNCS 1241, pp. 220—242, 1997.

36. Kiselev, I. Aspect-Oriented Programming with AspectJ. Indianapolis, IN: SAMS, 2002.

37. Kruchten, Philippe. The Rational Unified Process. – Boston, MA: Addison-Wesley, 2003.

38. Safonov, Igor. Trust Engineering and Risk Management of Complex Systems. – Modeling and Analysis of Safety, Risk and Quality in Complex Systems. Proceedings of International Scientific School. – Saint-Petersburg: RAS, 2001.

39. Safonov, Igor. Aspect-Oriented Software Reliability Engineering. – Modeling and Analysis of Safety and Risk in Complex Systems. Proceedings of Third International Scientific School. – Saint-Petersburg: RAS, 2003.

40. Safonov, Igor, and Vadim Safonov. Forecasting and Planning of Corporate Business Activity and Data Processing for Optimal Trust Engineering and Risk Management. – Modeling and Analysis of Safety and Risk in Complex Systems. Proceedings of Forth International Scientific School. – Saint-Petersburg: RAS, 2004.

41. Safonov, Igor. Evolution Risks of Informational Technologies. – Ibid.

42. Safonov, Igor, and Vadim Safonov. Security Engineering and Patch Management for Safety of Information Technologies. – Thirteen Scientific-Technical Conference “Safety Systems” of the International Informatization Forum. – Moscow: IIA, 2004.

43. Safonov, Igor. Workflow Optimization for Safety Performance”. – Ibid.

44. Wake, William C. Refactoring Workbook. – Boston, MA: Addison-Wesley, 2004.