

Рефлексивное управление. (Международный симпозиум "Рефлексивные процессы и управление" 8-10 октября 2001 года, г. Москва) Сборник статей/Под редакцией В.Е.Лепского -М.: Изд-во "Институт психологии РАН", 2000.

## Рефлексирующие и мыслящие программные системы

Витяев Е.Е.

Институт Математики СО РАН, ЗАО "Новософт", [vityaev@math.nsc.ru](mailto:vityaev@math.nsc.ru)

Данная работа продолжает серию работ [1] по построению моделей программных систем на основе моделей мышления и рефлексии. Разрабатываемая модель строится с целью построения программных систем, в основании которых лежали бы новые принципы организации информационных процессов.

Основной проблемой при разработке программ, является «революционный» способ их создания, когда под немного измененные требования заказчика приходится практически заново создавать программный продукт. Кроме того, программные системы не подстраиваются под приемы работы пользователя, не обучаются новым приемам переработки информации, не планируют процесс обработки информации, не могут сами модифицировать постановку задачи и т.д. Все эти недостатки связаны с тем, что программные системы не рефлексируют над процессом своей работы и тем более не мыслят. В данной работе предлагается перенести некоторые знания из психологии, лингвистики, философии и математики, связанные с рефлексией и мышлением, на программные системы.

**Понятие задачи.** Известно [2], что в процессе целенаправленной деятельности Цель деятельности, определяется некоторой потребностью. Затем, в процессе принятия решений, она трансформируется в ряд промежуточных Результатов (подцелей), выполнение которых контролируется Акцептором Результатов Действий. Можно показать [3], что математической формализацией понятия Цель, является понятие Задача, введенное в [4]. Задача, в том числе математическая, имеет смысл только тогда, когда имеется критерий ее решения. В терминах Теории Функциональных Систем [2] это означает, что необходимым элементом целенаправленной деятельности является фиксация Результата деятельности. Математические задачи, как правило, определяются тем, что *дано*, и что *требуется* доказать. Решением математической задачи традиционно считается доказательство, построенное на начальных данных и не нарушающее правил математической логики.

С точки зрения программных систем, задача состоит в построении спецификации программной системы удовлетворяющей следующим критериям:

- полученная спецификация удовлетворяет всем требованиям заказчика;
- по этой спецификации автоматически синтезируется (например, дедуктивным синтезом) требуемая программная система;
- программная система по начальным данным должна получить требуемый результат;
- новые пожелания или изменения, идущие от заказчика могут быть легко внесены в спецификацию;

Последний пункт представляет наибольшие трудности для построения программных систем, в виду того, что приходится анализировать системы понятий, в которых мыслит заказчик. Кроме того, мы полагаем, что рефлексирующая и мыслящая программная система должна:

- настраиваться на пользователя и на его стиль работы;
- планировать оптимальный способ достижения Цели;
- модифицировать постановку задачи в случае невозможности достижения результата;
- автоматически учитывать и оптимизировать ресурсы.

Построение такой программной системы невозможно без хорошо разработанной и легко модифицируемой системы понятий.

**Система понятий.** Важными для нас свойствами системы понятий являются следующие:

- наличие системы базовых понятий – наиболее элементарных понятий, в терминах которых может быть определена большая часть остальных понятий. (Возможность построения таких «семантических пространств» исследовалась в [5]);
- иерархия понятий, например, по уровням абстрактного и конкретного;
- возможность изменения «видения», «точки зрения», «взгляда» на проблему (задачу) путем перехода из одной системы понятий в другую.

Задача, как это аргументируется в [4], формулируется не во всей системе понятий. Каждая задача выделяет из системы понятий свой фрагмент, в котором она формулируется и решается. Этот фрагмент образует «контекст» решения задачи. Этот контекст может варьироваться мышлением в процессе решения задачи путем «трансляции» данного фрагмента системы понятий как по уровням абстрактного и конкретного, так и изменением «видения» задачи путем перевода ее из одной системы понятий в другую.

**Мышление.** Основными функциями мышления являются следующие:

1. Движение по иерархии понятий (от абстрактного к конкретному и обратно);
2. Анализ и Синтез;
3. Изменение «видения» задачи (постановки) за счет изменения системы понятий.

Первая функция осуществляется за счет возможности движения («трансляции») по уровням абстракции системы понятий в процессе решения задачи.

Вторая функция применительно к программным системам может осуществляться за счет получения прямым и обратным выводом всех следствий из того, что *дано* и что *требуется* (анализ) на всех уровнях абстрактного и конкретного. Этот процесс сводит решение исходной задачи к серии подзадач, из решения которых, используя полученные выводы, будет следовать решение всей задачи (синтез).

Третья функция мышления осуществляется за счет изменения «видения» (постановки) задачи (одной из полученных в результате анализа задач) путем «трансляции» всех входящих в постановку задачи понятий из одной системы понятий в другую. В случае, когда задача формулируется системой квазитожеств, алгоритм «трансляции» этой задачи в другую систему понятий известен [6, стр.282].

**Рефлексия.** Рефлексия означает возможность анализа всего процесса решения задач, т.е. постоянного программного анализа самого процесса решения (протокола решения). Рефлексия включает обнаружение и поиск новых стратегий решения задач. Это определение может содержать регресс в бесконечность, если допустить возможность рефлексии над самим процессом рефлексии. Если этого не допускать, то потребуется предположить существование внешней неизменяемой программы, которая рефлексиирует над всеми остальными. Это ставит проблему погружения всех возможных уровней рефлексии в один язык спецификаций. Например, на естественном языке мы можем описывать рефлексии любого уровня. Примером формального языка, в котором возможно описание всех уровней рефлексии является  $\lambda$ -исчисление без типов. Можно ли в нем создавать спецификации мыслящих и рефлексиирующих программных систем остается неисследованным. Поэтому возникает задача разработки языка спецификаций, включающего все возможности работы с понятиями, мышлением и рефлексией.

## ЛИТЕРАТУРА

1. Ващенко В.В., Витяев Е.Е., Загоруйко Н.Г., Мальцев А.А., Непейвода Н.Н., Пальчунов Д.Е., Пыркин С.Г., Ткачев А.В. Информационные рефлексиирующие системы (в печати).
2. Анохин П.К. Принципиальные вопросы теории функциональных систем // Философские аспекты теории функциональных систем. М.: Наука, 1978. с. 49 - 106.
3. Витяев Е.Е. Целеполагание как принцип работы мозга // Модели когнитивных процессов (Выч. системы, 158), Труды ИМ СО РАН, Новосибирск, 1997, с.9-52.
4. Ершов Ю.Л., Самохвалов К.Ф. О новом подходе к философии математики // Структурный анализ символьных последовательностей (Выч. системы, 101), Новосибирск, 1984. с. 141-148.
5. В.Ф.Петренко Психосемантика сознания. МГУ, 1988, с.208.
6. А.И.Мальцев Алгебраические системы. «Наука», М., 1970, 392с.