

АРИЗ: от человека к компьютеру.

©Рубин М.С., Сысоев С.С., 2016

Ключевые слова: ТРИЗ, АРИЗ, алгоритм решения изобретательских задач, компьютерная программа, Compinno-TRIZ, супер-язык программирования.

1. Развитие алгоритмов решения изобретательских задач (АРИЗ).

Основная цель программного комплекса Compinno-TRIZ (Компас инноваций на основе инструментов ТРИЗ) состоит в создании автоматизированной системы постановки и решения изобретательских задач. В основе Compinno-TRIZ лежит идея объединения АРИЗ, функционального анализа и применения инструментов комплексного анализа систем [18].

История развития АРИЗ изложена в [6, 7].

Первый вариант АРИЗ изложен в работе Г.Альтшуллера и Р. Шапиро в первой публикации по ТРИЗ в 1956 г. [1]. Вот краткий перечень модификаций АРИЗ в разные годы.

АРИЗ-56 – Аналитическая, Оперативная и Синтетическая стадии (Альтшуллер, Шапиро)

АРИЗ-59 – введен ИКР, поиск в других отраслях техники (Альтшуллер)

АРИЗ-61 – прообраз приемов разрешения противоречий (Альтшуллер)

АРИЗ-62 – стадия выбора задачи (Альтшуллер)

АРИЗ-63 – прообраз таблицы применения приемов (Альтшуллер)

АРИЗ-64/65, АРИЗ-65 – 31 прием; таблица 16x16 (Альтшуллер)

АРИЗ-68 – стадия уточнения задачи, 35 приемов, таблица 32x32 (Альтшуллер)

АРИЗ-71, АРИЗ-71Б, АРИЗ-71В – оператор РВС, 40 приемов, таблица 39x39 (Альтшуллер)

АРИЗ-77 – веполь, стандарты, физические противоречия, физические эффекты, ММЧ (Альтшуллер)

АРИЗ-82 с модификациями – введено микро-ФП, таблица решения физических противоречий (Альтшуллер)

АРИЗ-85-В – ВПР, ресурсный ИКР, стандарты-76, нет таблицы применения приемов (Альтшуллер [2])

АРИЗ-СМВА-91 – это вариант АРИЗ, ориентированный на выполнение в форме компьютерной программы, с детальным описанием каждого шага, но программа создана не была (Злотин, Зусман [3, 4])

АРИЗ-91 – более детальный и точный, но сложный алгоритм, (коллектив авторов под руководством Литвина [5])

АРИП-2009ПТ – алгоритм решения инженерных проблем (Иванов)

АРИЗ-2010 в отличие от АРИЗ-85-В является многоуровневым: сначала используется самый простой вариант для решения задачи (логика АРИЗ),

если не удовлетворены решением, то средний (практический АРИЗ) и затем детальный. Шаги АРИЗ прописаны более детально, ФП рассматривается на нескольких уровнях, выстраивая причинно-следственную цепочку. Частично использованы отдельные части из АРИЗ-91 в виде приложения. (Петров) [10].

АРИЗ-Универсал 2010 – за основу взят АРИЗ-85-В, может применяться для технических и нетехнических систем, введен элепольный анализ и новая система стандартов, ИКР свойств и другие понятия и инструменты, введена логика парных шагов: «анализ-синтез» (Рубин [11])

АРИЗ-У-2014 – вариант АРИЗ-2010, адаптированный для реализации в виде компьютерной программы, задача описывается как комплекс полезных и вредных функций, использованы формальные алгоритмы перехода от шага к шагу, часть шагов АРИЗ выполняется автоматически [16].

В 2009 и в 2012 годах на конференциях Саммита разработчиков ТРИЗ были проведены исследования и дискуссии по созданию АРИЗ нового поколения. В частности, была проведена экспертная оценка трех новых на тот момент модификаций АРИЗ (В.Петрова, Г.Иванова, М.Рубина). На рис. 1 показано, что в целом оценки экспертов положительные, новые варианты АРИЗ (в том числе и АРИЗ-У-2010) могут использоваться в обучении и в практике решения изобретательских задач.



Рисунок 1. Экспертная оценка модификаций АРИЗ (процент положительных ответов независимых экспертов – Мастеров ТРИЗ) [9].

Основное противоречие развития современных модификаций АРИЗ связано с двумя взаимопротиворечащими друг другу требованиями:

- АРИЗ должен быть максимально формализован, все шаги должны быть детально расписаны, он должен включать максимальное количество

инструментов и информационных материалов, что приводит к сильному усложнению АРИЗ и увеличению необходимого времени на его освоение;

- АРИЗ должен быть максимально простым для изучения и использования, он должен требовать минимального времени на освоение.

Чаще всего сейчас стремятся идти по пути упрощения инструментов ТРИЗ и в первую очередь АРИЗ. Из него убираются стандарты и вспомогательный анализ, анализ противоречий, все сводится к 5-7 вопросам, которые уже больше напоминают метод контрольных вопросов, а не АРИЗ. В некоторых случаях весь ТРИЗ и АРИЗ сводят просто к использованию таблицы Альтшуллера¹.

Для удовлетворения противоречивых требований к АРИЗ необходимо использовать АРИЗ с максимальными функциональными возможностями, но очень простой для усвоения и применения пользователем. Таким является АРИЗ-У-2014. Достигается это за счет переноса основной логической и информационной нагрузки алгоритма на компьютерную программу, человеку остаются только самые важные действия и принятие окончательных решений. АРИЗ-У-2014 можно начинать использовать с минимальным уровнем подготовки по ТРИЗ. Это не значит, что после первой разобранной задачи по АРИЗ-У-2014 пользователь автоматически становится специалистом по ТРИЗ. АРИЗ-У-2014 дает подсказки и сам учит пользователя в процессе решения изобретательских задач. АРИЗ-У-2014 совмещает в себе и функции инструмента решения изобретательских задач, и функции обучения инструментам ТРИЗ, и функции выработки ТРИЗ-овского мышления, которое формируется в процессе многократного применения АРИЗ-У-2014, реализованного в программном комплексе Compinno-TRIZ.

2. Описание АРИЗ-У-2014 и его реализации в программном комплексе Compinno-TRIZ.

В настоящее время в состав комплекса Compinno-TRIZ включены:

- АРИЗ-У-2014 – модернизированный АРИЗ Универсал 2010, позволяющий автоматизировать выполнение ключевых шагов АРИЗ

- Универсальная система стандартов на решение изобретательских задач - 2010 с уточненной редакцией от 14.06.2014 года, основанная на элепольном анализе

- Компонентный, структурный и функциональный анализ

- Формирование обобщенной функции для перехода к ФОП

- Фрагменты указателя физических эффектов

- Инструменты постановки и переформулировки задач, выявления вторичных задач.

Некоторые шаги АРИЗ-У-2014 в Compinno-TRIZ делаются полностью автоматически, некоторые шаги выполняются на основе выбора из

¹ См. например, на <http://www.firefly-consulting.com/triz/> таблица Альтшуллера представлена как инструмент прорывных инноваций.

предлагаемого списка вариантов, часть шагов требует ответа на однозначно поставленные вопросы.

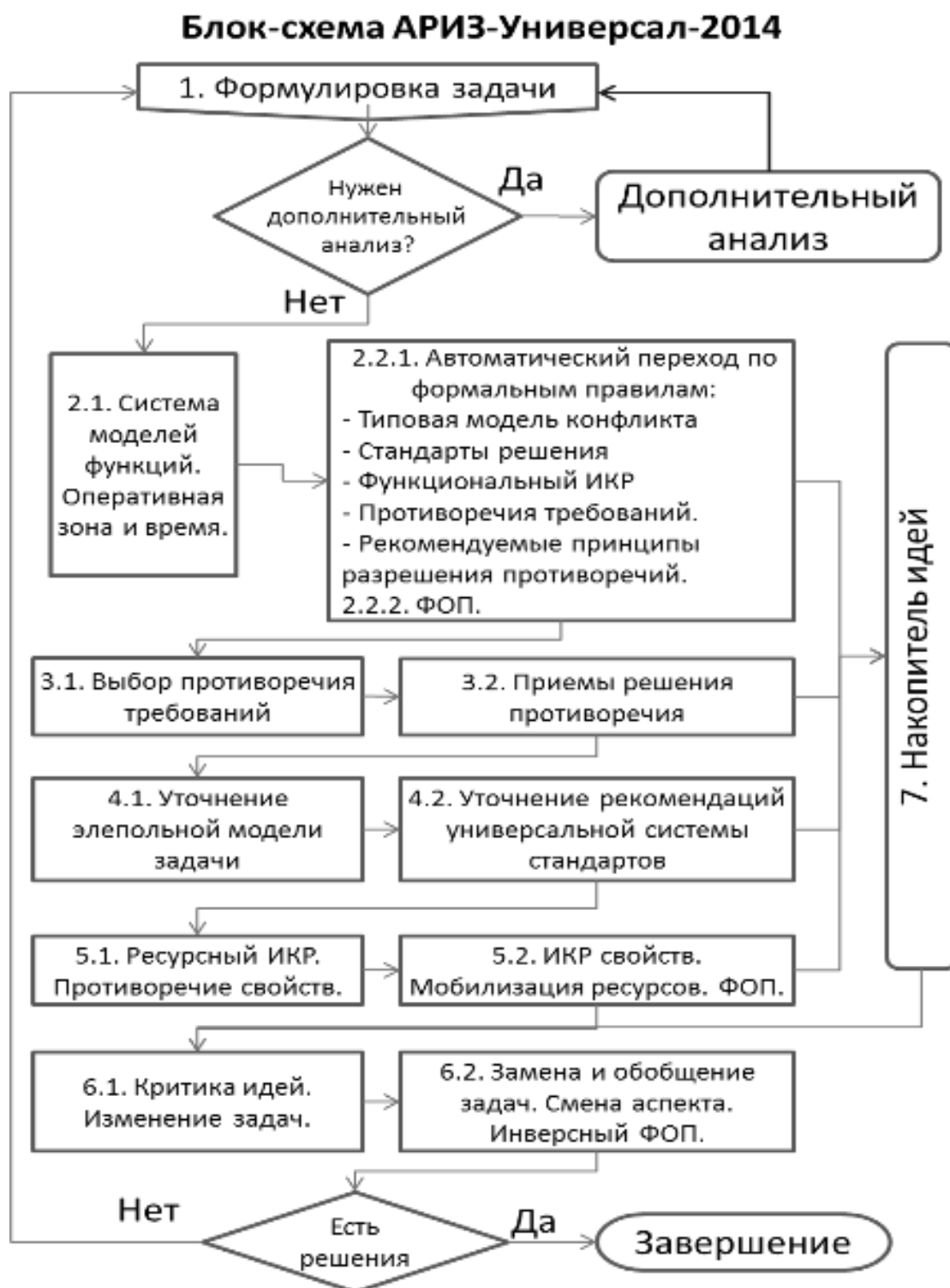


Рисунок 2. Блок схема АРИЗ-У-2014.

На рис. 1 представлена блок схема АРИЗ-У-2014. С подробным описанием алгоритма можно познакомиться в [16]. Программный комплекс Compinno-TRIZ разработан на языке Python и развернут в инфраструктуре Google AppEngine. Интерфейс пользователя реализован в виде веб-

приложения, доступного по адресу <http://ariz-2010.appspot.com/>. Более детальное описание комплекса Compinno-TRIZ изложено в [18].

Мы познакомимся с логикой работы комплекса Compinno-TRIZ на примере разбора конкретной изобретательской задачи из области программирования.

Разбор задачи о сортировке массива. Задача. Массив, содержащий, например, целые числа, можно отсортировать методом пузырька. Однако время выполнения алгоритма растет квадратично от длины массива, то есть для достаточно длинных массивов это время становится недопустимо большим. Можно создать новый алгоритм сортировки, но на это также потребуется много времени. Создание нового метода “с нуля” является трудной задачей.

Как сократить время сортировки длинного массива, не создавая новый алгоритм сортировки? [13].

Первый цикл анализа задачи.

Шаг 1. Вводятся компоненты системы и их характеристики (параметры):

Метод пузырька (скорость работы);

Новый алгоритм сортировки (Скорость работы, сложность создания);

Массив данных (Длина массива, упорядоченность, время сортировки).

В компонентах всегда есть X-элемент и X-поле.

Шаг 2. Формируем систему функций для этой задачи и ограничения:

Субъект	Глагол	Изменение	Параметр	Объект	Тип функции
X-элемент	Ускоряет	Увеличивает	Скорость работы	Метод пузырька	Неосуществимая, недостаточная
Метод пузырька	Сортирует	Увеличивает	Время сортировки	Массив данных	Вредная
Метод пузырька	Сортирует	Увеличивает	Упорядоченность полезная	Массив данных	Полезная

Ограничения: Нельзя создавать новый алгоритм сортировки данных.

Примечание 1. Ввод параметра (характеристики) не является обязательным, но он важен с точки зрения формулировки противоречия и понимания сути задачи. Например, одна и та же функция (Метод пузырька Сортирует Массив данных) по параметру увеличение упорядоченности полезная, а по параметру увеличение времени сортировки – вредная.

Примечание 2. В эту таблицу можно вносить не только функции, но и какие-то характеристики взаимоотношений Субъекта, и Объекта (принадлежит, владеет, является родственником и т.д.).

Шаг 3. Если в качестве проблемной на предыдущем шаге выбрать первую функцию (X-элемент ускоряет (увеличивает скорость работы) Метод пузырька (неосуществимая функция), то для такой ситуации алгоритмы Compinno-TRIZ автоматически по правилам свертывания или по системе стандартов [14, 17] дают рекомендации по моделям решения задачи: стандарт 1.1 Создание элеполевой структуры (новой системы) – синтез элеполя, вводя недостающие элементы. В линии стандартов введения элементов рекомендуется использовать для введения уже имеющиеся элементы или их

модификации. В линии дробления и динамизации рекомендуется выделить отдельный элемент, который рассматривается как целое, разделить его на две части (би-элемент) и соединить их между собой полем взаимодействия.

Примечание 3. На этом шаге программа может предложить изменить выполнение 2-го шага, если не выбрана проблемная функция, или перечень функций не содержит проблемные ситуации. Может потребоваться введение новых объектов на шаге 1 для формулировки этих функций.

Шаг 4. На этом шаге автоматически дается формулировка Функционального ИКР: Массив данных (или можно выбрать любой другой объект из списка на шаге 2) САМ ускоряет (увеличивает скорость работы) объекта Метод пузырька. Предлагается провести информационный поиск по поисковому образу «ускоряет Метод пузырька» и его модификациям.

Шаг 5. На шаге формулировки противоречия требований от пользователя требуется заполнить строчку «ЕСЛИ...». Например, можно ввести «создать новый метод сортировки». Остальные части противоречия и его противоположности формируются почти автоматически. Из этих двух формулировок требуется выбрать одно: ЕСЛИ не создать новый метод сортировки ТО не Нарушается ограничение Нельзя создавать новый алгоритм сортировки массива данных НО не Выполняется функция X-элемент ускоряет (увеличивает скорость работы) объекта Метод пузырька.

Примечание 4. На этом шаге тоже может потребоваться вернуться к шагу 2 для введения недостающих формулировок функций (полезных, вредных или недостаточных). Это происходит, если не удается сформулировать противоречие требований на основе уже введенных функций.

Шаг 6. На этом шаге предлагается определить, какими принципами разрешения противоречий можно воспользоваться. В нашем случае кажется, что не подходит принцип разделения требований во времени (полезная и вредная функция реализуется одновременно), в пространстве разделить требования тоже не удастся, в отношениях тоже не видно возможностей решить противоречие. Использовать физико-химические преобразования и фазовые переходы для математической задачи тоже нельзя. Остается один принцип: разрешения противоречивых требований системным переходом (в надсистеме, подсистеме, в структуре). Для этого принципа рекомендуются следующие приемы:

- | | |
|----------------------------|------------------------------|
| 1. Дробления | 27. Дешевая недолговечность |
| 5. Объединения | взамен дорогой долговечности |
| 6. Универсальности | 33. Однородности |
| 13. Наоборот | 42. Принцип |
| 16. Частичного или | многоступенчатого действия |
| избыточного действия | 43. Применение пены |
| 22. Обратить вред в пользу | 45. Би-принцип |
| 23. Обратной связи | 50. Принцип самоорганизации |
| 25. Самообслуживания | |

Шаг 7. На шаге «Матрица» используется таблица применения приемов разрешения противоречий. Можно выбрать две пары срок и столбцов: «09 Скорость - 36 Сложность устройства» и вторая пара «09 Скорость - 32 Удобство изготовления». На этом же шаге выбираем вариант использования сокращенного варианта таблицы и приемов [15], так как мы имеем дело с нематериальным объектом. Можно было бы выбрать и полный вариант таблицы.

Примечание 5. В сокращенном варианте таблицы Альтишуллера исключаются неподходящие для нематериальных систем параметры (вес, площадь, энергия и др.) и приемы (изменение физико-химических параметров, замена механической схемы и др.)

Шаг 8. На основе такого выбора предлагается список из пяти приемов:

1. Принцип дробления (а. Разделить объект на независимые части; б. Выполнить объект разборным; в. Увеличить степень дробления объекта)

4. Принцип асимметрии

10. Принцип предварительного действия (а. Заранее выполнить требуемое действие; б. Заранее расставить объекты так, чтобы они могли вступить в действие без затрат времени на доставку и с наиболее удобного места)

13. Принцип наоборот

34. Принцип отброса и регенерации частей

С учетом рекомендаций шага 6 получаем список рекомендованных приемов: дробления (рекомендован дважды), предварительного действия, би-принцип и другие приемы.

Шаг 9. Элеполи и применение стандартов. На этом шаге можно уточнить выбор стандартов с учетом проведенного к этому шага анализа. Мы пропустим в данном разборе этот шаг.

Шаг 10. На этом шаге формулируется противоречие свойств:

Параметр «Длина массива» объекта «Массив данных» должен быть короткий и Должен быть НЕ короткий и ИКР свойств:

Элемент «Массив данных» с «Длина массива» короткий САМ должен обеспечить свойство длинный.

Для этого пользователь:

- выбирает из предлагаемого списка объект, из-за свойства которого возникает противоречие (Массив данных)
- выбирает из списка параметр или характеристику этого объекта, которая приводит к противоречию (Длина массива)
- вписывает противоречивые значения этого параметра или характеристики (короткий, не короткий, длинный).

Шаг 11. На этом шаге автоматически даются формулировки:

- Ресурсный ИКР: X-ресурс (из ресурсов системы) на месте элемента «Массив данных», сохраняя его характеристику 'короткий', должен САМ в течение времени действия в пределах зоны действия обеспечивать свойство длинный.

- Микро-ресурсный ИКР: Микроэлементы из ресурсов системы на месте элемента «Массив данных», сохраняя его характеристику 'короткий ', должны САМИ в течение времени действия в пределах зоны действия обеспечивать свойство длинный.

Шаг 12. «Оценка решений и формулировка вторичных задач». Это просто контрольные вопросы и рекомендации по определению вторичных задач или переформулировке исходной задачи.

По ходу анализа основные формулировки и идеи записываются в Накопитель идей. В нашем случае мы получаем следующий образ решения задачи.

Элемент «Массив данных» с «Длина массива короткий» САМ должен обеспечить свойство длинный.

«Массив данных» САМ ускоряет (увеличивает скорость работы) объекта «Метод пузырька».

Нужно использовать принцип системного перехода (к подсистемам) и приемы дробления, предварительного действия, перехода к би-системе.

Необходимо модифицировать имеющиеся элементы (Массив данных). Целое разделить на две части (би-элемент) и соединить их между собой полем взаимодействия.

Этот образ полностью соответствует контрольному решению задачи.

Второй цикл анализа задачи.

На шаге 2 возможен другой вариант выбора проблемной функции. Можно выбрать вторую функцию (вредную): «Метод пузырька» Сортирует (Увеличивает Время сортировки) «Массив данных» (Вредная).

В этом случае на шаге 3 система стандартов [17] рекомендует два стандарта: 1.1 Создание элепольной структуры (новой системы) (как и на первом цикле анализа) и 1.2.1 Устранение вредных связей заменой, изменением или дополнением элементов: задачу решают заменой (изменением) одного из элементов или введением третьего элемента.

Использовать в качестве ЭЗ видоизменения Э1 и/или Э2;

Рекомендуется также использовать линию введения элементов.

Рекомендации свертываний:

- Нет необходимости в функции Метод пузырька сортирует (увеличивает упорядоченность объекта) Массив данных;

- Вместо Метод пузырька функцию сортирует выполняет другой ресурсный элемент;

- Элемент Массив данных сам выполняет эту функцию.

То есть в последней формулировке для ускорения сортировки рекомендуется использовать сам Массив данных.

Мои проекты

ariz-2010.appspot.com/project/?name=Задача%20о%20сортировке%20массива&cycle=2

Compinno-TRIZ Задача о сортировке массива АРИЗ-У-2014 Стандарты Физ.эффекты Идеи

АРИЗ-У-2014

Описание

Компоненты

Функции

Модели решения

ИКР-Противоречия

Матрица

Приемы

Элементы

ИКР 2

Ресурсы

Задачи

Идеи

Назад Дальше

Предварительные рекомендации по направлениям возможного решения

Задача: Требуется устранить вредную функцию: Метод пузырька сортирует (увеличивает время сортировки объекта) Массив данных

Рекомендации функционального анализа

Исключить необходимость выполнения одной из функций (свертывание):

1. Нет необходимости в функции Метод пузырька сортирует (увеличивает упорядоченность объекта) Массив данных. Вместо Метод пузырька функцию сортирует выполняет другой ресурсный элемент, Элемент Массив данных сам выполняет эту функцию.
2. Нет необходимости в функции X-элемент ускоряет (увеличивает скорость работы объекта) Метод пузырька. Вместо X-элемент функцию ускоряет выполняет другой ресурсный элемент, Элемент Метод пузырька сам выполняет эту функцию.

1.2.1 Устранение вредных связей заменой, изменением или дополнением элементов

Если между двумя элементами в элеполе возникают сопряженные – полезное и вредное – действия, задачу решают заменой (изменением) одного из элементов или введением третьего элемента. Замена (или изменение) Э2 должно быть

Рисунок 3. Compinno-TRIZ. Задача о сортировке массива. Цикл 2. Шаг "Модели решения".

Шаг 4, цикл 2. Предлагается три формулировки функционального ИКР.

а) «Массив данных» (или другой элемент из компонентов системы) САМ устраняет сортирует (увеличивает время сортировки) объекта Массив данных.

б) Массив данных (или другой элемент из названных компонентов системы, кроме субъекта вредной функции) САМ сортирует (увеличивает упорядоченность объекта) Массив данных.

в) Массив данных (или другой элемент из названных компонентов системы и надсистемы) САМ исключает необходимость выполнения ускоряет (увеличивает скорость работы объекта) Метод пузырька. Выбираем формулировку б) и получаем такую же формулировку функционального ИКР, как и на первом цикле анализа: Массив данных САМ ускоряет (увеличивает скорость работы) объекта Метод пузырька. Соответственно и все остальные шаги выполняются так же, как на первом цикле.

Контрольное решение. Согласно принципу дробления и принципу предварительного действия, нужно до начала сортировки разбить массив на части (например, на две равные части). При этом каждый из двух новых массивов будет меньше начального. Затем нужно применить алгоритм сортировки к каждому массиву по отдельности, после чего соединить отдельные упорядоченные массивы в один. При этом, так как время соединения упорядоченных массивов линейно зависит от их длины, а время сортировки пузырьком - квадратично, то общее время сортировки начального массива будет уменьшено. Причем чем длиннее массив, тем больший выигрыш во времени будет получен [13].

3. Использование программного комплекса Compinno-TRIZ при обучении студентов и инженеров.

Комплекс Compinno-TRIZ сейчас все еще находится в стадии формирования. Это авторский, научно-исследовательский, некоммерческий проект. Все это время нас больше интересовали задачи, связанные с определением принципиальной выполнимости поставленных целей: автоматизировать работу с АРИЗ, минимизировать время освоения инструментов ТРИЗ, не снижая уровня качества используемых инструментов решения изобретательских задач и инновационного проектирования.

Сейчас нам хотелось бы уделить больше внимания методикам применения Compinno-TRIZ в процессе обучения ТРИЗ. Мы приведем несколько наиболее типичных примеров из нашего еще небольшого опыта использования этого комплекса в обучении ТРИЗ студентов, инженеров и ученых разных специальностей.

3.1. Обучение ТРИЗ школьников.

Группа школьников в Санкт-Петербурге в объединении «Мастерская изобретателя» проходит обучение по ТРИЗ с 1-го класса в течении 5 лет и освоила основные инструменты ТРИЗ, основы курса РТВ. Они используют при решении изобретательских задач понятия противоречия требований, противоречия свойств, ИКР, ресурсы, приемы разрешения противоречий, ММЧ. Для более полного освоения инструментов ТРИЗ (АРИЗ, стандарты, элпольный анализ, функциональный анализ) с 6-го класса в 2016-2017 учебном году они будут использовать программный комплекс Compinno-TRIZ. Будут отработаны методики применения комплекса Compinno-TRIZ при обучении школьников и разработаны рекомендации для развития этого комплекса при обучении школьников.

3.2. При обучении студентов. Санкт-Петербургский Государственный Университет, Математико-механический факультет, Кафедра информатики.

Примерная программа обучения основам ТРИЗ:

Форма и содержание занятий	Продолжительность
Вводная лекция по основным разделам ТРИЗ	3 часов
Демонстрация компьютерной реализации АРИЗ и совместное решение учебной изобретательской задачи.	2 часов
Самостоятельное решение 4-5 учебных изобретательских задач (домашние задания)	Примерно 5 часов самостоятельной работы
Индивидуальный разбор домашних заданий	1-2 часа

Четверо из этой группы студентов выполнили после краткого обучения выпускные дипломные работы по теме ТРИЗ в программировании² (руководители Одинцов И., Рубин М.) и с отличием защитили эти работы. ... Использование компьютерной реализации АРИЗ позволило за короткое время показать студентам логику работы АРИЗ, на практике показать, как работают основные инструменты ТРИЗ при решении изобретательских задач не только в технических, но и в нетехнических областях. Разбор задач по АРИЗ стал продолжением обучения инструментам ТРИЗ.

3.3. При обучении магистров. Санкт-Петербургский Государственный Университет, Математико-механический факультет, Магистратура.

С 2012 года для магистров математико-механического факультета СПбГУ читается спецкурс по ТРИЗ в информационных технологиях продолжительностью 1 семестр и нагрузкой 1 зачетная единица (32 академических часа).

Слушателями курса являются, как правило, опытные инженеры, уже имеющие степень бакалавра в области информатики. Особенности преподавания ТРИЗ специалистам в своей предметной области неоднократно обсуждались разработчиками теории. Инженера трудно заставить правильно применять инструменты ТРИЗ, потому что он уже «знает, как надо». При использовании программного комплекса Compinno-TRIZ (далее – ПК) подобные трудности часто удается преодолеть по следующим причинам:

1. В ПК жестко заложены основные этапы и шаги алгоритма. У пользователя отсутствует возможность продолжить содержательную работу при пропуске важных шагов (таких, как компонентный или анализ функций).
2. Практически на каждом шаге от пользователя требуется переформулировка задачи в рамках логики этого шага. Эта особенность вызывает основные трудности при использовании АРИЗ, которых обычно пытаются избежать и применить инструмент

² <http://www.temm.ru/ru/section.php?docId=4538>

неправильно. Логика ПК не позволяет пользователю «сжутьничать» и вынуждает формулировать задачу так, как этого требует АРИЗ.

Результатом применения Compinno-TRIZ часто является полное или частичное устранение инерции мышления в области решаемой задачи, а также большое количество свежих идей, возникающих при «борьбе с формулировками». Кроме того, у пользователя не возникает задачи самостоятельного поиска по информационному фонду, поскольку соответствующие задаче стандарты, формулировки и методы формируются автоматически на основе формализованного описания задачи.

Идея формализации описания задачи для автоматического применения инструментов ТРИЗ может иметь развитие в виде формального языка (декларативного языка программирования) – TRIZ-L (см. ниже).

В качестве примера использования формального подхода при описании задачи приведем формулировки, полученные при рассмотрении изобретательской ситуации «Противодействие терроризму» на семинаре с магистрами в 2016-м году.

В результате функционального анализа была сформулирована вредная функция:

Террористы нападают на **Уязвимые объекты** (уменьшают параметр **Качество**).

Для того, чтобы вредная функция стала причиной функционального конфликта, объекты **Террористы** и **Уязвимые объекты** должны быть представлены в качестве участников полезных функций. С уязвимыми объектами трудностей не возникло, а вот с террористами пришлось поломать голову. Проблема заключается в том, что если террористы не участвуют в полезной функции, то функциональный анализ предписывает просто вычеркнуть их из системы. Невозможность этого обусловлена тем, что террористы на самом деле не являются для системы внешним явлением, а представляют ее часть – часть того множества (люди), которое страдает от их действий. На основе этих рассуждений была сформулирована полезная функция:

Террористы составляют объект **Люди** (стабилизируют параметр **Количество**).

После этого функциональный конфликт был подтвержден системой. Автоматически сгенерированные рекомендации по свертыванию содержали, например, следующие формулировки:

1. Нет необходимости в функции **террористы составляют (стабилизирует количество объекта) люди**. Вместо **террористы** функцию **составляют** выполняет другой ресурсный элемент; Элемент **люди** сам выполняет эту функцию.

2. Нет необходимости в функции **уязвимые объекты сохраняет (стабилизирует количество объекта) люди**. Вместо **уязвимые объекты** функцию **сохраняет** выполняет другой ресурсный элемент; Элемент **люди** сам выполняет эту функцию.

3. Нет необходимости в функции **армия защищает (стабилизирует качество объекта) уязвимые объекты**. Вместо **армия** функцию **защищает** выполняет другой ресурсный элемент; Элемент **уязвимые объекты** сам выполняет эту функцию.

На основании этих формулировок можно сформулировать ряд вторичных задач:

1. Как предотвратить превращение людей в террористов?
2. Как сделать уязвимые для атак объекты неинтересными для этих атак?
3. Как обеспечить уязвимые для атак объекты функциональностью самостоятельного противодействия атакам?

Третья подзадача активно решается в настоящее время с помощью разъяснительных мер для населения («что делать, если вами обнаружен подозрительный пакет или сумка»), установкой систем видеонаблюдения и пр.

Нам кажется, что детальный разбор всех трех задач с помощью АРИЗ мог бы дать много новых интересных решений.

3.4. При обучении преподавателей, инженеров и ученых. Семинар в Уральском Федеральном Университете в 2014 году (преподаватели Рубин М., Герасимов О.). 40-часовой базовый семинар по ТРИЗ. При обучении использовался АРИЗ-У-2010, Универсальная система стандартов на решение изобретательских задач. Для сравнения слушателям была предоставлена возможность входа в АРИЗ через формулировку противоречия требований (как в АРИЗ-85-В, АРИЗ-2010 и многих других модификациях АРИЗ) и другой вариант – вход в анализ через формулировку системы функций, как это делается в АРИЗ-У-2014.

Несколько задач: о дрессировщике львов, о скифах, о снайперской винтовке (задачи 7-9 в приложении к статье) и другие.

По мнению слушателей семинара вход в анализ изобретательской ситуации через формулировку системы функций проще и удобнее, позволяет более четко сформулировать суть изобретательской задачи.

Можно выделить особенности проведения обучения при использовании программного комплекса Compinno-TRIZ по сравнению с обычными занятиями по ТРИЗ:

- естественно, требуется соответствующее техническое обеспечение: компьютеры с доступом в Интернет, желательно проекционный аппарат и экран, для совместного разбора хода анализа;
- навыки выделения основных объектов, входящих в задачу с описанием их параметров и характеристик (перечень объектов обычно не вызывает затруднений, а с описанием параметров и характеристик объектов могут возникать сложности, требуется тренинг)
- формулировка моделей функций в формате «Субъект – действие – Объект» (требуется тренинг для правильной формулировки функций)
- обычно требуется тренинг для умения формулировать действие в формате параметр объекта и его изменение (увеличение-уменьшение, изменение-стабилизация, измерение). Например, увеличивать температуру воды (нагреть воду); стабилизировать расположение картины (повесить картину на стене) и т.д.
- умение так подбирать систему функций и выбирать проблемную функцию, чтобы формировалось противоречие требований.
- элементарные навыки работы с комплексом Compinno-TRIZ и интерпретацией получаемых рекомендаций.

4. О концепция языка программирования для описания изобретательских ситуаций.

Формирование интеллектуального языка программирования (супер-языка [20]) для создания алгоритмов и программ, содержащих в своем изначальном описании противоречия. В частности, имеется возможность переходить от моделей функций к набору стандартов на разрешение выявленных противоречий и далее возможен переход к паттернам программирования, как это описано в [14]. То можно построить формальные алгоритмы, осуществляющие автоматический переход от описания проблемной ситуации с противоречиями к структуре программы, которая может реализовать решение для такой изобретательской ситуации.

Можно дать краткое представление о концепции языка описания изобретательских ситуаций TRIZ-L. Язык описания изобретательских

ситуаций TRIZ-L служит для формализации следующих аспектов административного противоречия:

1. Функциональный конфликт. Необходимы директивы для объявления:
 - a. Объектов и их параметров.
 - b. Полей и их параметров.
 - c. Направленного графа отношений между объектами вида (объект_1 -> отношение -> объект_2).
 - d. Функций (субъект, параметр объекта, вид изменения, характер функции).
2. Противоречие требований. Необходимы директивы для объявления:
 - a. Объектов (см. пункт 1)
 - b. Действий, влияющих на параметры объектов (действие, улучшение параметра_1, ухудшение параметра_2).
 - c. Выбранного противоречия.
 - d. Противоречие свойств. По аналогии с пунктом 2, только параметр и объект один.
1. ИКР. Требуются директивы для объявления:
 - a. Ресурсов
 - b. Ограничений
 - c. ИКР (параметр объекта = ..., при ограничениях....)

Описание ситуации на языке TRIZ-L может служить входными данными для формальной обработки с использованием инструментов ТРИЗ:

1. Функциональный анализ
2. Элеполюсный, Вепольный анализ
3. Матрица Альтшуллера
4. Принципы разрешения противоречий (во времени, в пространстве и др.)
5. Ресурсный анализ
6. РТВ (метод числовой оси, ИКР, морфологический анализ, метод фантограмм и др.)

5. Выводы и заключение

1. АРИЗ-У-2014 и его реализация в программном комплексе Compinno-TRIZ может успешно применяться при решении изобретательских задач и при обучении на занятиях по ТРИЗ.

2. Комплекс Compinno-TRIZ позволяет формулировать и решать изобретательские задачи не только в технике, но и в нетехнических областях.

3. При формулировке изобретательских задач в форме комплекса моделей функций возникает возможности формализации противоречий требований, рекомендуемых стандартов и задач на свертывание, при формулировке ИКР и на других шагах АРИЗ. При формулировке могут использоваться описания не только модели функций, но и описание взаимоотношений между объектами иного вида.

4. Логика, заложенная в АРИЗ-У-2014, делает крайне сложным неправильную работу с АРИЗ, подсказывает пользователю корректные формулировки, снимает инерцию мышления, уточняет суть исходной задачи. Комплекс Compino-TRIZ является одновременно и учебным пособием, так как содержит в себе методические рекомендации и информационный фонд, необходимый для решения изобретательских задач.

5. Разрабатываемый в рамках развития комплекса Compino-TRIZ формальный аппарат ТРИЗ позволяет ставить цели создания концепции супер-языка программирования TRIZ-L для описания изобретательских ситуаций и автоматического формирования решений имеющихся в них изобретательских задач.

Литература

1. Альтшуллер Г.С., Шапиро Р.Б. Психология изобретательского творчества. - Вопросы психологии, 1956, № 6, с. 37-49. <http://www.altshuller.ru>
2. Альтшуллер Г.С. АРИЗ - значит победа. Алгоритм решения изобретательских задач АРИЗ-85-В.- Правила игры без правил / Сост. А.Б.Селюцкий. - Петрозаводск: Карелия, 1989.-280 с. - (Техника - молодежь - творчество), с. 11-50.
3. Злотин Б.Л., Зусман А.В. АРИЗ-СМВА-91. <http://triz-summit.ru/ru/205253/203840/204230/204231/204232/>
4. Злотин Б.Л., Зусман А.В. Проблемы развития АРИЗ, Журнал ТРИЗ, 3.1.1992, стр. 41 <http://triz-summit.ru/file.php/id/f300293-file-original.pdf>
5. АРИЗ-91, МУНТР, Литвин С.С. и др. Санкт-Петербург, 1997 г. <http://triz-summit.ru/ru/205253/203840/204230/204231/204295/>
6. Альтшуллер Г.С. История развития АРИЗ (конспект). Журнал ТРИЗ, 3.1.1992, стр. 38 <http://triz-summit.ru/file.php/id/f300293-file-original.pdf>
7. Петров В.М. История развития алгоритма решения изобретательских задач – АРИЗ. Издание 2-е, Тель-Авив, 2008. <http://triz-summit.ru/file.php/id/f302322-file-original.pdf>
8. В.М. Цуриков «Проект «Изобретающая машина» - интеллектуальная среда поддержки инженерной деятельности», Журнал ТРИЗ, 2.1.91., стр. 17 Ассоциация ТРИЗ, г. Обнинск, 1991 г. <http://triz-summit.ru/file.php/id/f300291-file-original.pdf>

9. Материалы конференции Саммита разработчиков ТРИЗ 2012. Статьи по теме "АРИЗ нового поколения" и «ТРИЗ в информационных системах», 2012, Санкт-Петербург <http://triz-summit.ru/ru/confer/TDS-2012/>
10. Петров В.М. АРИЗ-2010. <http://triz-summit.ru/ru/205253/203840/204230/205699/>
11. Рубин М.С., АРИЗ-Универсал-2010. Версия от 15.07.2012. <http://triz-summit.ru/ru/confer/TDS-2012/205266/205416/205301/>
12. Рубин М.С., Кирдин А.Н., АРИЗ-У-2010 - программная реализация, Санкт-Петербург, 2012 г. <http://www.temm.ru/ru/section.php?docId=4589>
13. Струсь Глеб Игоревич, Постановка и решение изобретательских задач в программировании на основе методов ТРИЗ. Дипломная работа, Санкт-Петербургский Государственный Университет, Математико-механический факультет, Кафедра информатики, 2010 г. Научный руководитель: Одинцов И.О., Рубин М.С. <http://www.temm.ru/ru/section.php?docId=4538>
14. Зиненко Ольга Игоревна, Систематизация и анализ паттернов проектирования на основе стандартов теории решения изобретательских задач, Санкт-Петербургский Государственный Университет, Математико-механический факультет, Кафедра информатики, 2010 г. Научный руководитель: Одинцов И.О., Рубин М.С. <http://www.temm.ru/ru/section.php?docId=4538>
15. Рубин М. С. Основы ТРИЗ. Применение ТРИЗ в программных и информационных системах: Учебное пособие. – СПб: АТМ Книга, 2011.
16. Рубин М.С. О разработке АРИЗ-Универсал-2014, TRIZFest 2014, материалы конференции, Прага, 2014 год.
17. Рубин М.С., Универсальная система стандартов на решение изобретательских задач – 2010. Версия от 18.02.2012. <http://triz-summit.ru/ru/confer/TDS-2012/205266/205416/205422/>
18. Рубин М.С., Сысоев С.С., Compino-TRIZ – программный комплекс постановки и решения изобретательских задач. 2014 <http://triz-summit.ru/file.php/id/f300069-file-original.pdf>
19. Одинцов И.О., Рубин М.С. Опыт применения методов ТРИЗ для повышение эффективности разработки ПО. Международная конференция "Разработка ПО 2009" 28-29 октября 2009 года, Москва <http://www.temm.ru/ru/section.php?docId=4419>
20. Алгинина Е. В. Проблемы создания супер-языка программирования // Научное сообщество студентов XXI столетия. Технические науки: сб. ст. по мат. XV междунар. студ. науч.-практ. конф. № 15. URL: [http://sibac.info/archive/technic/9\(12\).pdf](http://sibac.info/archive/technic/9(12).pdf) (дата обращения: 07.05.2016)
-

Приложение.

Примеры учебных изобретательских задач, которые решались при помощи АРИЗ-У-2014 и комплекса Compinno-TRIZ.

Приведем некоторые примеры учебных изобретательских задач, на которых опробовался Комплекс Compinno-TRIZ.

1. Задача о режимах редактора геометрических фигур (Струсь Г.)

В редакторе геометрических фигур (в основном – ломаных линий и многоугольников) существуют несколько режимов редактирования. Два из них отвечают за изменение формы редактируемых фигур. Первый режим – режим перемещения точек – позволяет с помощью мыши передвинуть любую вершину фигуры в заданную точку. Второй режим позволяет добавить к контуру новую вершину (щелчком мыши ребро разбивается на две части). При использовании такого редактора возникает неудобство: приходится постоянно переключаться из одного режима в другой.

Как сделать использование этих двух режимов удобным?

2. Задача о защите общедоступной программы (Струсь Г.)

Достаточно сложная и уникальная программа расчета была доступна сотрудникам института в виде файла в машинных кодах. Были опубликованы также результаты работы этой программы: исходные данные, результаты расчетов.

Как сделать так, чтобы доступной всем программой мог пользоваться только сам автор этой программы?

3. Задача о программе вычисления произвольного полинома (Струсь Г.)

В программе, предназначенной для вычисления значения произвольного полинома, имеется текстовое поле для ввода. Введенная строка затем проверяется на правильность (действительно ли пользователь ввел полином, а не произвольную строку), а затем распознается (создается модель полинома для его последующего вычисления). Однако, написание такой программы вызывает трудности, возникает большое количество ошибок. Кроме того, усложняется структура выполнения программы. Необходимо упростить программу.

4. Задача о редакторе документов (Зиненко О.)

Рассмотрим редактор документов, который допускает встраивание в документ графических объектов. Затраты на создание некоторых таких объектов, например больших растровых изображений, могут быть весьма значительны. Но документ должен открываться быстро, поэтому следует избегать создания всех «тяжелых» объектов на стадии открытия (да и вообще это излишне, поскольку не все они будут видны одновременно). Разумно создавать «тяжелые» объекты по требованию. Это означает «когда изображение становится видимым». Как это сделать?

5. Задача о производительности программы в ГРИД-среде (Зиненко О.)

Была написана программа, работающая в ГРИД-среде. В процессе использования выяснилось, что скорость работы программы в ГРИД соизмерима со скоростью работы на одном компьютере. Хотя предполагается, что скорость работы в распределенной среде должна быть значительно выше. Как быть?

6. Задача об оптимизации кода программы (Зиненко О.)

Процесс разработки приложения велся N лет, к текущему варианту постепенно добавлялась новая функциональность. Анализ выявил, что 20% всего кода составляет полезный (функциональный) код, а 80% - неоптимизированный связующий код. Нужно оптимизировать код программы.

7. Задача о голове дрессировщика в пасти льва (Викентьев И.)

Один из самых захватывающих номеров, которые демонстрируют дрессировщики львов - это когда лев широко раскрывает свою пасть, а дрессировщик помещает туда свою голову. Номер очень и очень опасный, ведь даже самый послушный лев остается хищником, как бы хорошо он ни был дрессирован. Как же дрессировщикам удается показать этот трюк?

8. Задача о лучной атаке скифов (Рубин М.).

В глубокой древности главным преимуществом кочевых племен скифов в бою были лошади. Однако использовать их в лучной атаке оказалось очень сложно. Скифы шли на сближение с противником, поднимали лук со стрелкой, и, естественно, отпускали поводья. В этот момент лошади, получив свободу, оборачивались, и скифы оказывались спинами к врагу. Как быть скифам?

9. Задача о колебаниях воздуха у снайперской винтовки (Овчинникова В.)

Даже небольшие движения теплого воздуха, возникающие из-за конвекционных потоков нагретого ствола снайперской винтовки с глушителем, делают работу снайпера практически невозможной. Нагреться ствол может не только от стрельбы, но и просто из-за солнечных лучей. Снайперский прицел многократно усиливает изменение изображения даже при малейшем движении воздуха. Как обеспечить нормальные условия для стрельбы снайпера?

10. Задача о контроле качества угля на Апатитской ТЭЦ (Рубин М.).

11. Задача о маркировке «Дедал» в Древней Греции (Мурашковский Ю.) и другие учебные и реальные изобретательские задачи.

Для проверки АРИЗ-У-2014 анализировались и классические задачи: «Задача о макете парашюта для контроля вихреобразования», «Задача о молниеотводе», «Задача о шлаке», «Задача о перекачке сжатого газа из транспортного баллона» и многие другие учебные изобретательские задачи.