ПРИЧИННО-СЛЕДСТВЕННЫЙ АНАЛИЗ

В.Н. Болотовский,
Санкт-Петербург

В работе «Прикладная проблемология» было показано, что для решения проблем необходимо сперва из проблемы сформулировать задачи, пригодные для решения методами ТРИЗ. Сделать это можно, построив причинно-следственные цепочки.
В этой работе будут подробно рассмотрены свойства и особенности самих причинно-следственных цепочек (далее ПСЦ).

1. Элементы ПСЦ

Из построения и общих соображений видно, что ПСЦ состоят из узлов и ветвей . Какой же смысл имеют узлы и ветви ПСЦ?
Сначала определим терминологию.
Объект – это некая сущность реального мира, без ограничений на его физическую природу. Объекты в процессе анализа представляются с помощью ***моделей***.
Модель – набор ***параметров***, с помощью которых можно описать объект с достаточной точностью в данной ситуации, и множества их возможных ***значений***.
Параметр (свойство, характеристика) - отличительная особенность объекта, определяющая характер его взаимодействия с другими объектами или явлениями.
Значение параметра – элемент множества возможных значений, который принимает параметр в рассматриваемый момент.
Состояние объекта – ситуация, при которой определенный(е) параметр(ы) объекта имеют значения из определенного подмножества множества возможных значений.

Примеры:
Объект – вода, параметр – температура, множество возможных значений – от 0 ^{o}C до 100 ^{o}C, состояния – до 12 ^{o}C – холодная, от 12 ^{o}C до 50 ^{o}C – тёплая, больше 50 ^{o}C – горячая.
Объект – обои; параметр – цвет; множество возможных значений – индиго, бежевый, бордо, светло-зелёный; состояния – бежевый, светло-зелёный – светлые; индиго, бордо - тёмные.
Состояния могут определяться достаточно произвольно, из соображений того, чтобы объект в разных состояниях по-разному взаимодействовал с другими объектами. Состояния должны полностью перекрывать все множество возможных значений.
Карта состояний – множество возможных значений параметра объекта, разбитое на подмножества, характеризующие определенные состояния.
Дополнительное состояние – фиктивное состояние, при котором параметр

может принимать значения не входящие в множество значений, принимаемых параметром в рассматриваемом состоянии.

Во многих случаях, нас интересует какое-то определенное состояние объекта по какому-то параметру (в общем случае состояние может быть многомерным, по нескольким параметрам). Тогда удобно определить состояния по данному параметру, как А и НЕ А (дополнительное к А состояние, где имеются ввиду не противоположные значения, а все остальные). Например, если температура воды нас интересует с точки зрения заваривания чая, то она имеет состояния – горячая и не горячая, а если с точки зрения купания, то соответственно тёплая и не тёплая. Во втором случае, вода с температурой +3 ^{o}C и с температурой +80 ^{o}C имеет одинаковое состояние.

1.1. Узлы ПСЦ

Что же находится в узлах ПСЦ?

Рассмотрим узлы А и В. Из определения «А должно [быть] В» видно, что А – это объект, а В – это состояние объекта. Из второго плеча проблемы «А не должно/ не может [быть] В видно, что объект А имеет два состояния – В и не В. В рассматриваемый момент объект А находится в состоянии не В, но его нужно перевести в состояние В.

Что же такое узлы С и D? Эти узлы представляют собой объекты, находящиеся в неком состоянии. То есть одновременно объект и его состояние. С этой точки зрения логично узлы А и В объединить в один, и считать его проблемным звеном.

Далее.

По смыслу самой ПСЦ, каждый объект, находясь в определенном состоянии, влечет за собой.. что? Некое состояние другого объекта (или того же самого, но по другому параметру). Тогда логично в качестве узлов ПСЦ рассматривать карту состояний объекта. То есть для каждого узла ПСЦ необходимо задать объект, параметр и значение (поскольку параметр и значение определяют состояние, то вместо значения можно указать состояние). Например: вода – температура < +20°C/< +20°C, или проще: вода – холодная/тёплая. Состояние воды влечет за собой определенное состояние объекта «купание», по параметру «комфорт»: Холодная вода – некомфортно, теплая – комфортно.

1.2. Ветви ПСЦ

Что же представляют собой ветви ПСЦ?

В функциональном анализе объекты связаны друг с другом посредством функций. Можно ли рассматривать ветви ПСЦ, как функции?

Оказывается можно, но для этого нужно рассматривать функции с точки зрения информационно-параметрического подхода.

В классическом функциональном анализе функцией является воздействие одного объекта на другой, вызывающее изменение в последнем. В информационно-параметрическом подходе, функция – это изменение значения параметра объекта (изделия), вследствие определенного значения параметра у

другого объекта (инструмента). А это как раз то, что мы наблюдаем в ПСЦ. Только с той разницей, что воздействие передаётся не от значения параметра к значению параметра, а от состояния к состоянию (то есть к области значений).

Следственная цепочка является, так сказать, «обратно-функциональной». То есть она определяет какое состояние какого объекта приводит к рассматриваемому. Ветвь следственной цепочки – та же функция, но с точки зрения изделия. Поэтому формально, и причинные и следственные цепочки – суть одно и тоже, и могу быть объединены в Причинно-Следственную Цепочку. Отличие причинной ветви от следственной заключается в направлении функционального воздействия: от проблемного звена (причинная ветвь), или к проблемному звену (следственная ветвь).

Теперь определим, что же такое Проблемное звено.

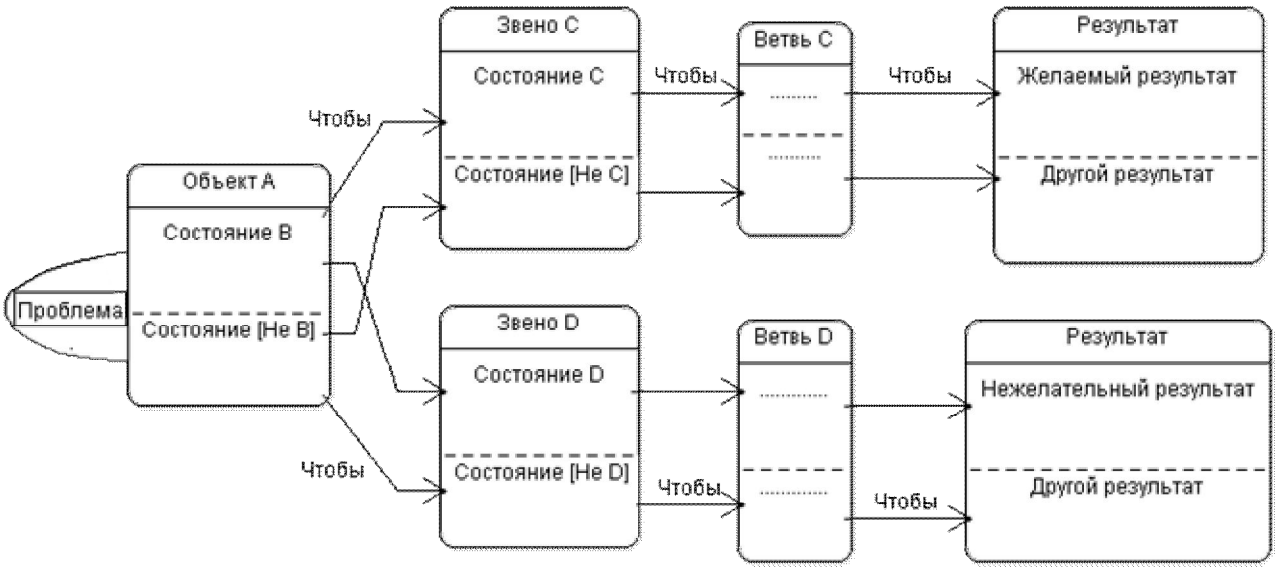
1.3.Проблемное звено.

Причинно-следственные цепочки строятся не ради искусства. Построение ПСЦ начинается с проблемной ситуации, в которой А должно [быть] В, чтобы С, но А не должно (или не может) [быть] В, чтобы (или потому что) D. Это формализация Административного Противоречия. В ПСЦ это утверждение «складывается» в три звена, где одно звено – С, другое D, а третье составное, в котором А – объект, В – одно из его состояний, включающее в себя и параметр, и область его значений. Соответственно, не В – это дополнительная область значений, то есть дополнительное состояние. Это означает, что объект А должен находится в двух взаимоисключающих состояниях.

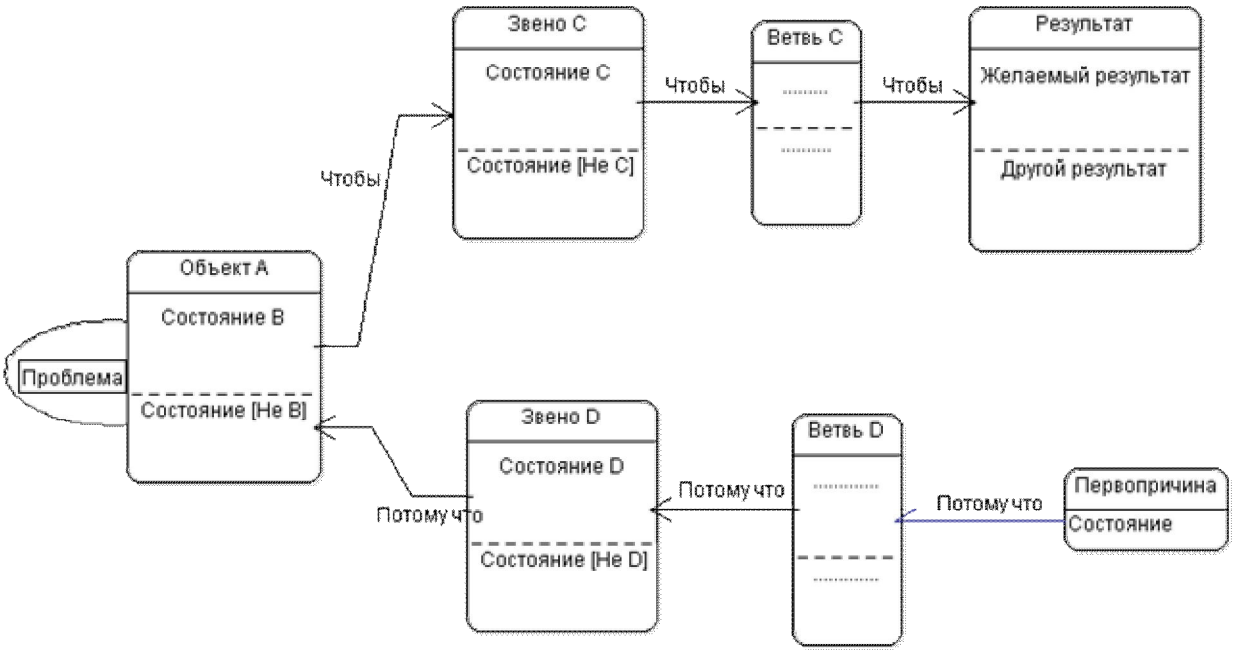
Это звено в ПСЦ и является проблемным звеном. В том смысле, что построенные звенья ПСЦ логически (и физически) вытекают друг из друга, и «сходятся» на проблемном звене в разных его состояниях.

Схематически это можно изобразить так:

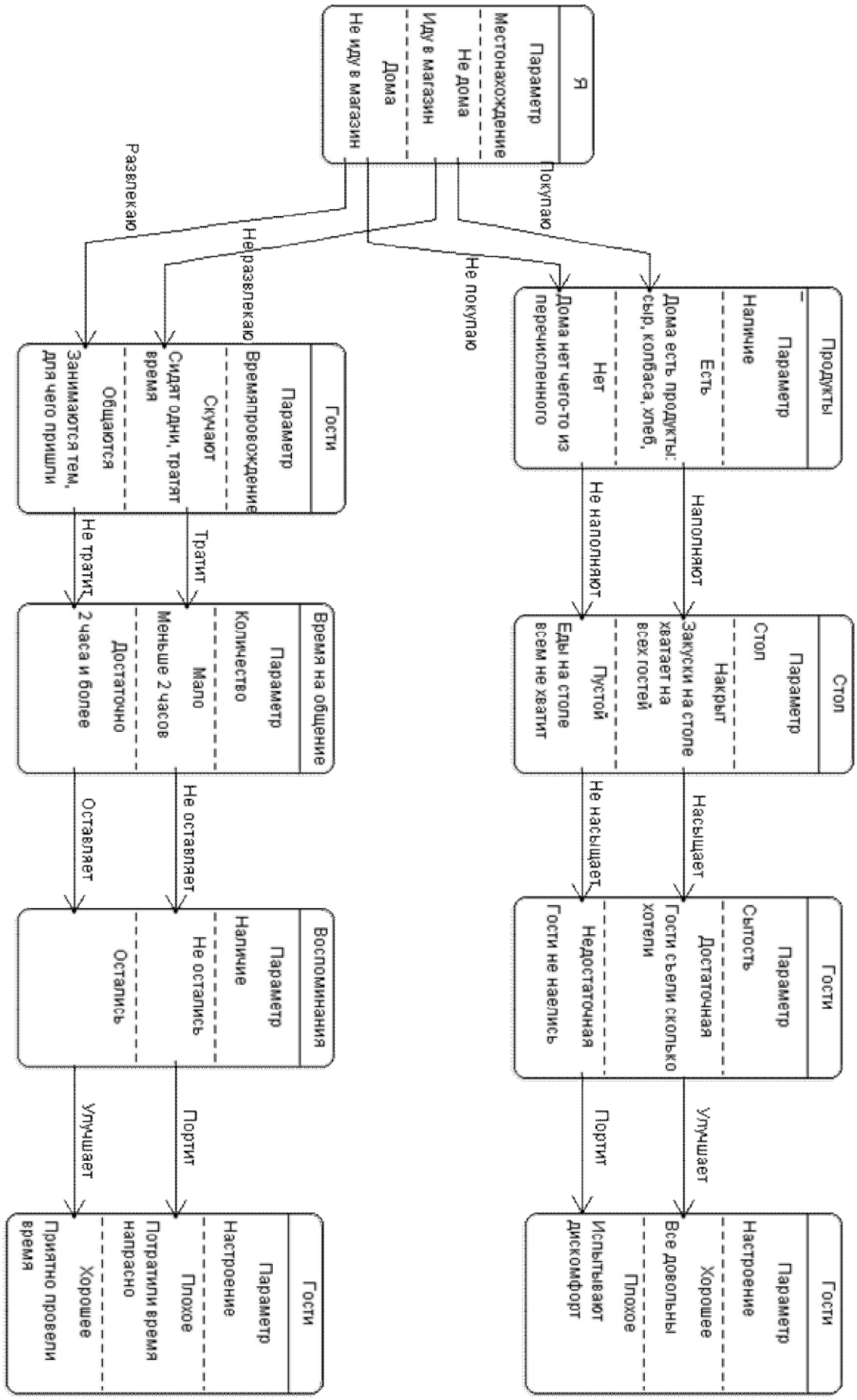
Для следственно-следственной цепочки



Для причинно-следственной цепочки



Если рассмотреть ветви (стрелки) как функции объектов, а состояния как области значений определенного параметра, то можно представить пример ПЦС в следующем виде:


Я
Параметр
Местонахождение
Не дома
Иду в магазин
Дома
Не иду в магазин
Покупаю
Не покупаю
Развлекаю
Неразвлекаю
Продукты
Параметр
Наличие
Есть
Дома есть продукты: сыр, колбаса, хлеб,
Нет
Дома нет чего-то из перечисленного
Наполняют
Не наполняют
Стол
Параметр
Стол
Накрыт
Закуски на столе хватает на всех гостей
Пустой
Еды на столе всем не хватит
Насыщает
Не насыщает
Гости
Параметр
Сытость
Достаточная
Гости съели сколько хотели
Недостаточная
Гости не наелись
Улучшает
Портит
Гости
Параметр
Настроение
Хорошее
Все довольны
Плохое
Испытывают дискомфорт
Гости
Параметр
Времяпровождение
Скучают
Сидят одни, тратят время
Общаются
Занимаются тем, для чего пришли
Тратит
Не тратит
Время на общение
Параметр
Количество
Мало
Меньше 2 часов
Достаточно
2 часа и более
Не оставляет
Оставляет
Воспоминания
Параметр
Наличие
Не остались
Остались
Портит
Улучшает
Гости
Параметр
Настроение
Плохое
Потратили время напрасно
Хорошее
Приятно провели время

:

1.4. Связь между состоянием объекта и его функцией.

Из приведенной выше схемы можно сделать определенные выводы.

1) Функция, выполняемая объектом, определяется его состоянием, то есть областью значений некоего параметра.
2) Верно и обратное: если объект выполняет некую функцию, значит, существует параметр, и область его значений (состояние объекта), которые обуславливают данную функцию
3) Если мы хотим обеспечить выполнение некой функции, для этого достаточно перевести инструмент в состояние, которое эту функцию обеспечивает.
4) Если мы хотим избавиться от вредной функции, достаточно вывести инструмент из такого состояния. Вывести его можно с помощью функции другого инструмента
5) Функция объекта **может быть направлена на тот же объект** (состояние объекта по одному параметру изменяет состояние объекта по другому параметру).
6) Причинно-следственная цепочка – это каскад состояний различных объектов, связанных функциональными воздействиями.
7) Состояние объекта, которое обеспечивает (посредством функции) другое состояние, также обеспечивает (посредством ПСЦ) все состояния, следующие по цепочке.
8) Состояние объекта определяется не только состоянием воздействующего на него объекта, но и состояниями всех предшествующих в цепочке объектов
9) Одно и то же состояние может обеспечивать несколько функций.
10) Состояние может определяться несколькими функциями (их конъюнкцией и/или дизъюнкцией).

Последнее утверждение на схеме не отображено, и его следует оговорить отдельно.

Привести объект в определенное состояние, может как одна функция, так и совокупность нескольких функций разных объектов, или разных состояний одного объекта. (Напомню, что узлами ПСЦ являются не сами объекты, а их состояния. Например «скатерть белая» и «скатерть квадратная» это разные независимые узлы).
Причем эти состояния могут влиять на объект-изделие, как все вместе, так и по отдельности, либо в комбинации этих вариантов. На языке логической алгебры это называется «логическое умножение, И», когда искомое состояние достигается лишь при условии существования всех предшествующих звеньев, и «логическое сложение, ИЛИ», когда достаточно существование хотя бы одного из условий. Схематически это можно отобразить так:

2. Использование ПСЦ

Рассмотрим некоторые особенности ПСЦ, благодаря которым этот механизм может найти практическое применение.

В п.3.3 были показаны две схемы проблемных участков ПСЦ, для следственно-следственной цепочки и причинно-следственной цепочки. Эти типы ПСЦ имеют некие общие особенности, и каждая свою специфику.

2.1.Общие особенности ПСЦ

Обобщая вышеизложенное, можно заметить следующее.

Первое:

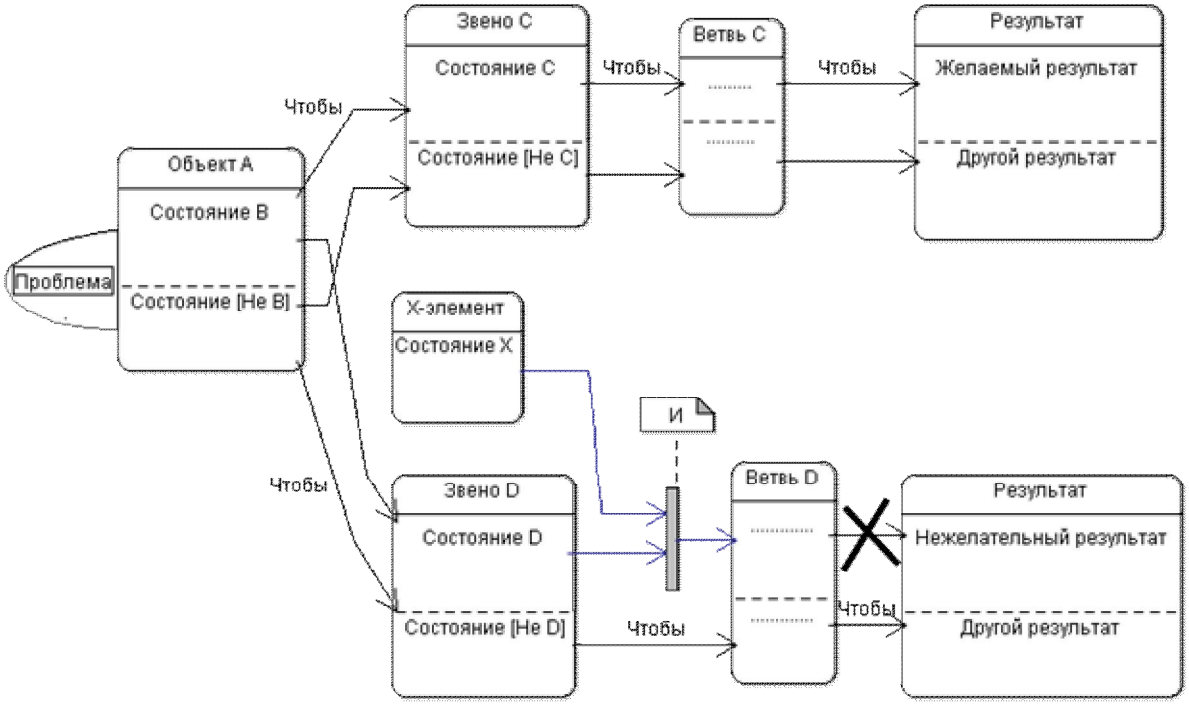
Проблемное звено и прилегающие к нему звенья ПСЦ являют собой видимую часть конфликта. И существует большое искушение решать его именно в этом месте. Но на самом деле конфликт имеет более глобальный характер, и конфликтуют между собой крайние звенья цепочки, а не те, которые находятся рядом с проблемным звеном.

Второе:

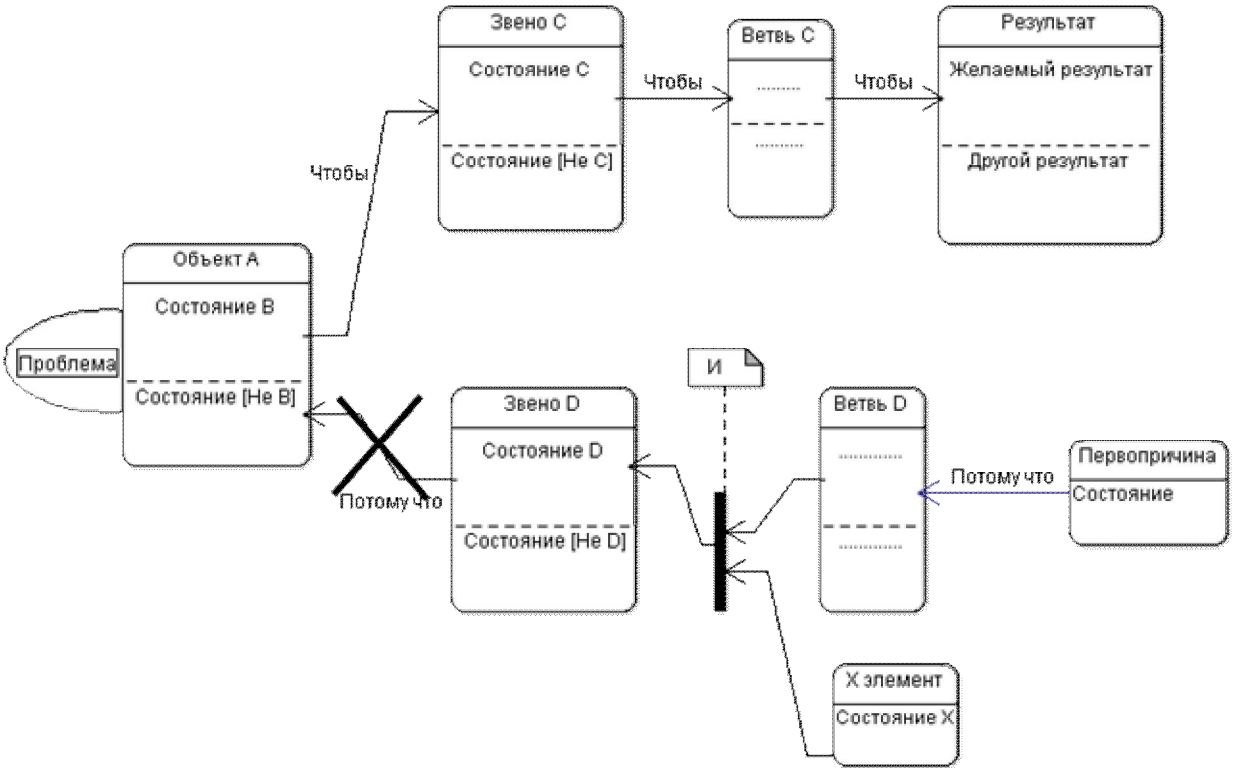
Функциональный поток, в какую сторону бы он ни был направлен, прерывается весь по ходу потока при удалении одного из звеньев. Соответственно проблема может быть разрешима на любом из звеньев, посредством устранения всей цепочки, приводящей к проблемному звену. Существует большая вероятность того, что на каком-либо звене ПСЦ решение видимой проблемы гораздо дешевле, очевиднее и менее трудозатратно. Но оно не так очевидно при попытках решить видимую часть проблемы. Примером может служить известная задача про перемешивание клевера и люцерны.

Третье:

Поскольку узлы ПСЦ можно логически складывать и логически умножать, то есть возможность прерывать функциональный поток с помощью «логических вентилей» - привнесенных узлов, которые легко управляемы, но могут закрывать или открывать функциональный поток. Этот метод широко используется в цифровой схемотехнике чуть ли не с момента её появления. Подробнее механизм показан на схеме:



Или:



Подобные вентили можно вставлять в любое звено нежелательной ветви ПСЦ. Это позволяет сформулировать целый каскад задач, решив которые, можно разрешить исходное противоречие. Примеры таких каскадов приведены в п.п. 2.1-2.3.

Еще одно полезное применение ПСЦ заключается в том, что с их помощью можно увидеть систему. В классическом ФСА сначала принято определить систему, после чего строить все структурные и функциональные связи, среди которых множество несущественных, а некоторые важные элементы оказываются вне системы. В этом есть свои плюсы, так как среди, казалось бы, несущественных элементов можно отыскать ресурс. Но есть и недостатки – такие, как «замусоренность» системы посторонними элементами и связями.

С помощью ПСЦ возможен другой подход к построению системы: сначала определяется проблемное звено, потом определяется вид ПСЦ, и строятся две ветви ПСЦ. Те объекты, которые вошли в узлы ПСЦ и составляют систему. Более того, ПСЦ сразу показывает и границы системы и функциональные связи между объектами. При таком подходе система, возможно не будет полной, но будет включать в себя все объекты, влияющие на возникновение проблемы.

ПСЦ также позволяют формулировать задачи, для решения их по АРИЗ. Поскольку на ПСЦ отображаются объекты, функции, параметры (свойства) и их значения, то с помощью ПСЦ можно сформулировать задачи для первого, второго и даже третьего шага АРИЗ.

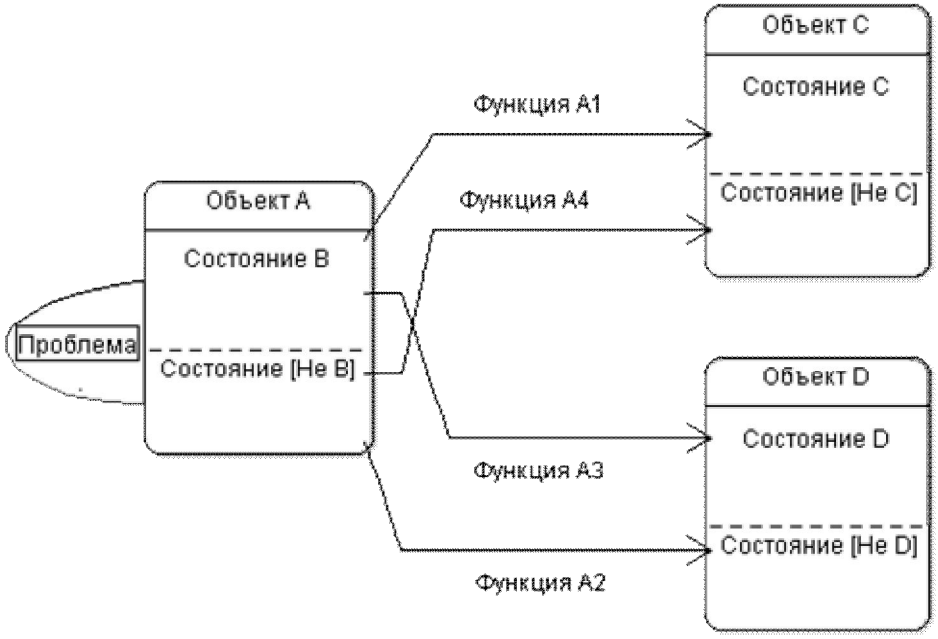
Рассмотрим эти возможности отдельно для разных видов ПСЦ.

2.2. Следственно-следственные цепочки

Из схемы конфликта видно, что в первом случае объект А в состоянии В приведет к желаемому результату, но также приведет и к нежелательному результату,

а состояние [не B] позволяет избежать нежелательный результат, но не позволяет достигнуть желаемого.

«Снаружи» проблема видна только в объекте A. Учитывая, что стрелки ПСЦ по сути являются функциями объекта A, а состояние означает «параметр B имеет значение в области B′»,схема позволяет сразу сформулировать ТП и ФП. Для этого представим проблемный участок в следующем виде:



Формулировка ТП1 : Объект A в состоянии B выполняет функцию A1 над объектом C, что хорошо, но также выполняет функцию A3 над объектом D, что плохо.

Формулировка ТП2: Объект A в состоянии [не B] выполняет функцию A2 над объектом D, что хорошо, но при этом не выполняет функцию A1 (выполняет A2) над объектом D что плохо.

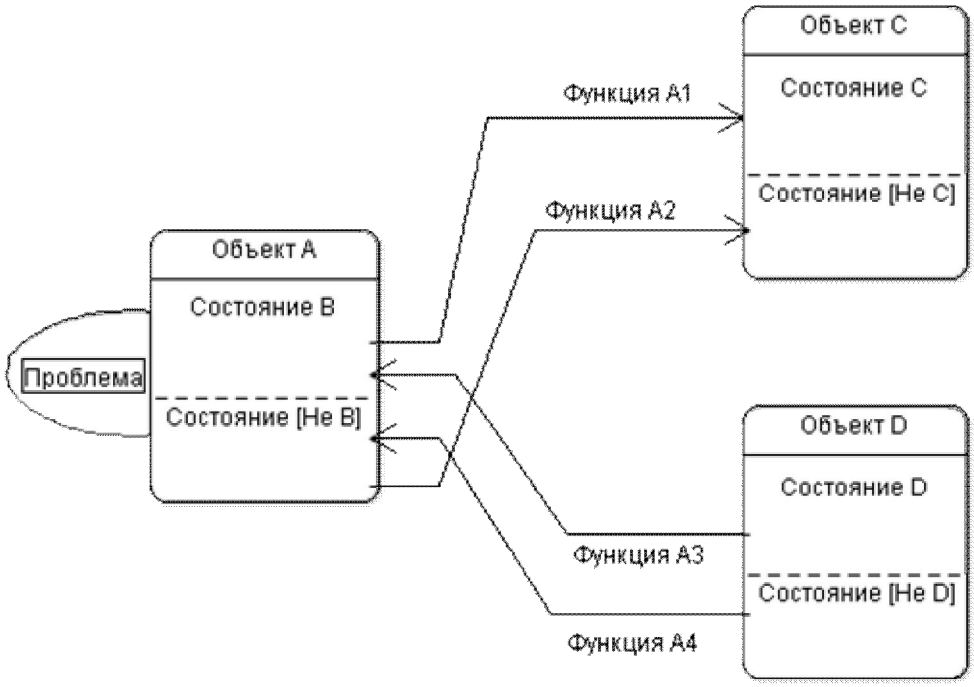
ФП: Параметр (свойство) B объекта A должен иметь значение в области B′ и должен иметь значение в области [не B′].

Обращаю внимание на то, что частица «не» используется не перед словом «должен», поскольку тут рассмотрен случай, когда у объекта только два состояния, перекрывающие всю область возможных значений.

Данный подход может быть обобщен на случай, когда объект A имеет не два, а большее число состояний. В этом случае, формулировка ФП должна включать фразу «должен находиться в состоянии B1, чтобы … и должен находиться в состоянии B2, чтобы … и должен находиться в состоянии B3, чтобы … Тут уже не подходит вариант «должен находиться в состоянии B, и НЕ должен находиться в состоянии B», поскольку «не B» в общем случае не означает «B1».

2.3.Причинно-следственные цепочки.

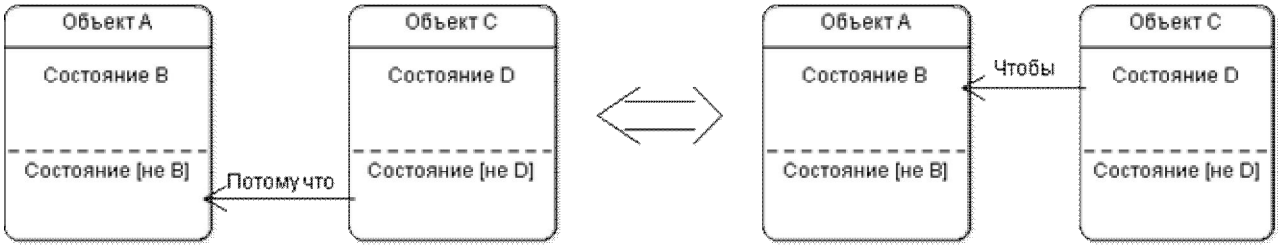
Для этого случая схема конфликта выглядит так:



Формулировка ТП1: Объект А должен выполнять функцию А1 по отношению к объекту С, но выполняет функцию А2, потому что объект D выполняет функцию А4.
Формулировка ТП2: Объект А выполняет функцию А2 потому что объект D выполняет функцию А4, и должен выполнять функцию А1 по отношению к объекту С.

ФП: Параметр (свойство) В объекта А должен иметь значение в области В′ но имеет значение в области [не В′].

В этом виде цепочек имеется одна особенность, для случая, когда состояний у объекта всего два. В цепочке можно провести такое преобразование:



То есть если объект-изделие имеет некое состояние **вследствие** определенного состояния инструмента, то **чтобы** перевести изделие в дополнительное состояние, инструмент должен быть переведен также в дополнительное состояние. Например: «вода *твердая*, **потому что** температура воздуха *ниже нуля*» эквивалентно «температура должна быть *не ниже нуля* **чтобы** вода была *не твердая*. Таким образом, можно перевести причинную ветвь ПСЦ в следственную, и наоборот. С

помощью этого преобразования можно «двигать» проблемное звено вдоль цепочки. Ведь в данном случае проблемным считается то звено, от которого отходят причинная и следственная ветви. Применив это преобразование, мы меняем одну ветвь с причинной на следственную и тем самым «сдвигаем» проблемное звено.

Таким образом, построив ПСЦ, можно отыскать решение видимой проблемы в любом месте ПСЦ, и найти такое решение, которое является совсем неочевидным при попытке решить проблему «в лоб».

Резюме

Итак, мы показали, что любая ситуация, которую в быту называют проблемой, имеет характерные особенности, и может быть разложена в причинно-следственную, либо в следственно-следственную цепочку. Построение ПСЦ к проблеме позволяет взглянуть на проблему шире, увидеть всю проблему целиком, а не только видимую её часть.

Исследуя построенную ПСЦ, можно найти проблемное звено, и сформулировать задачу, которую необходимо решить. Кроме формулировки задачи, ПСЦ позволяет сформулировать ТП, ФП, и составить систему, включив в неё только те элементы, которые влияют на решение искомой проблемы. Формулировки, полученные на этом шаге, позволяют применить АРИЗ, сразу на 1-ом, 2-ом или 3-ем шаге.

Используя свойство узлов, логически объединяться по «И» или по «ИЛИ», а также свойство ветвей разрушаться по ходу функционального потока, можно попытаться «отрубить» мешающую ветвь с помощью Х-элемента, либо другим способом. Для этого можно сформулировать задачи к каждому звену мешающей ветви, решение которых позволяют уничтожить ветвь целиком. Возможно, какая-нибудь из задач имеет очевидное решение, но совсем неочевидную связь с исходной задачей.

Применяя преобразование ПСЦ, можно «сдвигать» проблемное звено по цепочке, и формулировать задачи для АРИЗ к каждому из полученных проблемных звеньев.

Таким образом, нами получен самостоятельный алгоритм решения проблем, сопряженный с АРИЗ, и позволяющий значительно расширить поиск возможных решений, не перегружая его при этом заведомо тупиковыми направлениями.