

ВЕБ-ПРИЛОЖЕНИЕ ДЛЯ РИСОВАНИЯ ПРИЧИННО-СЛЕДСТВЕННЫХ ДИАГРАММ

В. А. Бычкова

*Белорусский государственный университет, г. Минск;
valerybychkova@tut.by; науч. рук. – К. Г. Апрохов*

Для качественного анализа сложных динамических систем используется инструментарий причинно-следственных диаграмм, способных объяснить сложную структуру взаимосвязей между переменными одной системы. Однако, большинство специализированных средств для рисования причинно-следственных диаграмм являются платными десктопными приложениями. Это затрудняет как создание диаграмм, так и совместную работу с ними.

Основным результатом работы является веб-приложение для рисования причинно-следственных диаграмм в браузере. В качестве основного инструмента для реализации сервиса была выбрана библиотека Joint.js.

Ключевые слова: системный анализ; динамическая система; причинно-следственные диаграммы; веб-разработка; JavaScript; Joint.js; SVG.

Потребность в принятии решений, для которых необходимо учесть обуславливающие их факторы, а также последствия их принятия, встречается во всех областях производства, экономики, хозяйства и социальной жизни. Сегодня реальность такова, что зачастую решения должны быть приняты в условиях сложной и быстроменяющейся окружающей среды. Применение распространенных традиционных аналитических методов к принятию решений в динамической среде достаточно ограничено и неэффективно, что подтверждает необходимость анализа обратной связи и эффекта запаздывания. Для анализа сложных динамических систем используется инструментарий причинно-следственных диаграмм.

Причинно-следственные диаграммы (диаграммы каузальных петель обратной связи – Causal Loop Diagrams, CLD) – один из методов анализа систем, который представляют собой эффективный инструмент для изучения структуры сложных систем. Причинно-следственные диаграммы просты в применении, не требуют существенной подготовки и специальных навыков моделирования. Важность причинно-следственных диаграмм состоит в их возможности наглядно показать этапы появления/развития проблемы, а кроме того обозначить пути ее решения.

Рассмотрим более подробно методологию построения CLD диаграмм.

Причинно-следственные диаграммы состоят из переменных и связей между ними с определенной полярностью (положительной или отрицательной). Связи между переменными изображаются стрелками.

Правила определения полярности связей [1]:

- Положительный контур обратной связи часто определяется так, что «начальное изменение любой переменной, в конечном счете, стимулирует далее самоизменение в первоначальном направлении». То есть, упрощенно: «чем больше – тем больше», «чем меньше – тем меньше».

- Для отрицательного контура обратной связи действует следующее классическое определение: «если отклик контура обратной связи на переменное изменение выступает против первоначального возмущения, то контур является отрицательным». То есть, упрощенно: «чем больше – тем меньше», и наоборот.

Например, продуктивность работы команды зависит от нескольких переменных. Применяя правило расстановки полярности, мы, соответственно, получаем следующую структуру системы: чем больше объем рутинной работы, тем меньше продуктивность (–); чем больше работников в офисе, тем выше продуктивность (+); чем сильнее боевой дух команды, тем выше продуктивность (+); наконец, чем больше необходимость в новых знаниях, тем меньше продуктивность, – следовательно, связь обратная (–) (рис. 1).

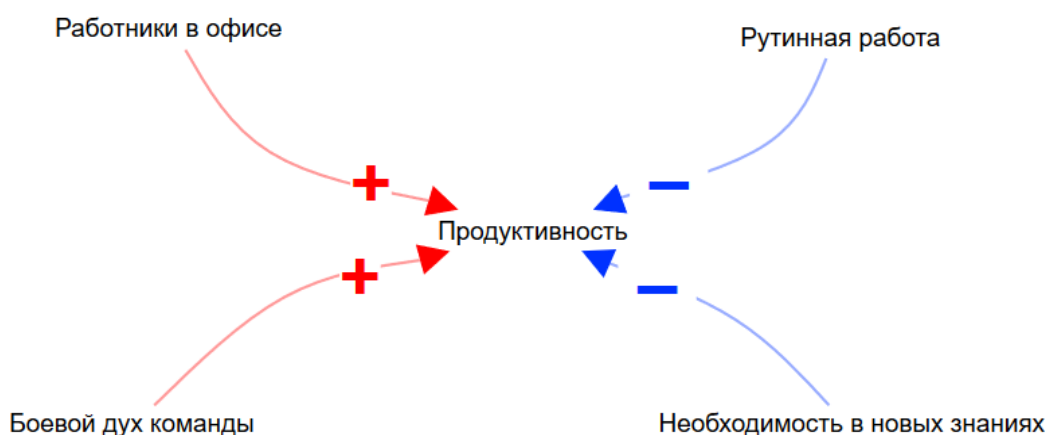


Рис. 1. Переменные, влияющие на продуктивность работы команды

Эффект, проявляемый только через какое-то время, отражается на диаграмме в виде двух линий на стрелке.

Во время анализа петель обратной связи исходят из количества отрицательных связей в петле. Позитивные петли обратной связи (помечаются как R от Reinforcing Loops) являются контурами с четным числом отрицательных связей («минусов»), а негативные петли обратной связи (B,

Balancing Loops) – контуры с нечетным числом «минусов». В соответствии с классическим определением, позитивный контур обратной связи определяется так: «начальное изменение любой переменной (в контуре), в конечном счете, стимулирует далее самоизменение в первоначальном направлении» [1].

Совокупность позитивных (усиливающих) и негативных (балансирующих) циклов обратной связи определяет поведение системы в целом [2].

Далее рассмотрим инструментарий необходимый для реализации веб-приложения.

Joint.js (JointJS) – это библиотека JavaScript для разработки диаграмм, с помощью SVG, CSS и HTML. Если стандартные механизмы рисования в Javascript, например, элемент CANVAS, полагаются на пиксели, то SVG – это векторная графика. Применение SVG позволяет создавать структуры с насыщенной графикой, обладающие анимацией и возможностями взаимодействия.

Библиотека содержит в себе различные элементы: прямоугольники, круги, овалы, изображение, а также произвольные фигуры, поддерживает возможность привязки элементов друг к другу, сериализацию/десериализацию данных в JSON, адаптацию под устройства с сенсорным дисплеем и многое другое. Для разработки диаграмм в данном проекте были задействованы следующие библиотеки:

- joint.js (библиотека для создания диаграмм);
- lodash.js (вспомогательная библиотека необходимая для joint.js);
- backbone.js (вспомогательная библиотека необходимая для joint.js);
- jquery-3.2.1.min.js (библиотека для упрощенной работы с объектами диаграмм);
- bootstrap.min.js (библиотека для создания диалоговых окон);
- pan-zoom.min.js (библиотека для реализации масштабируемости холста);
- bootstrap.min.css (фреймворк, содержащий HTML- и CSS-шаблоны кнопок, блоков, меток и прочих компонентов веб-интерфейса).

С помощью описанного выше инструментария реализовано веб-приложение следующего вида (рис. 2).

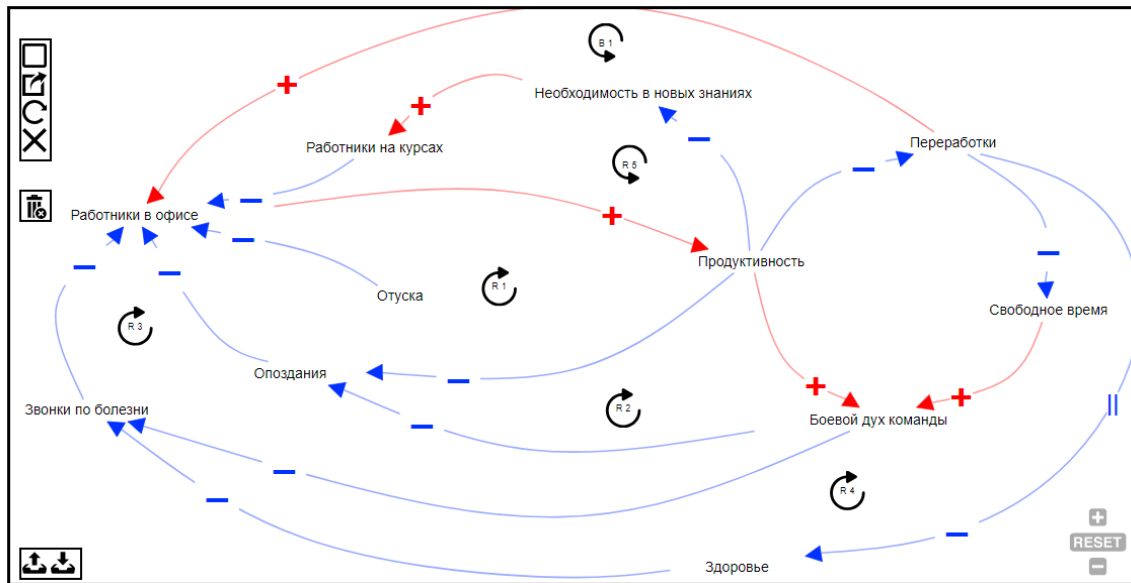


Рис. 2. Веб-приложение для рисования CLD диаграмм

Сервис позволяет создавать, редактировать и удалять различные объекты на холсте (переменные, связи, циклы), импортировать/экспортировать построенные диаграммы в формате JSON, масштабировать отображаемую текущую диаграмму под необходимые размеры экрана и/или диаграммы.

Управление диаграммами осуществляется посредством кнопок, комбинаций клавиш на клавиатуре, диалоговых окон и текстовых полей.

Библиографические ссылки

1. Каталевский Д. Ю. Основы имитационного моделирования и системного анализа в управлении : учеб. пособ. М., 2011.
2. Медоуз Д. Азбука системного мышления // пер. с англ. М., 2011.
3. Карпов Ю. Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. СПб., 2009.
4. JointJS: Visualize and interact with diagrams and graphs [Электронный ресурс]. URL: <https://www.jointjs.com/opensource> (дата обращения: 05.05.2018).
5. Systems Tools for Complex Health Systems: A Guide to Creating Causal Loop Diagrams [Электронный ресурс]. URL: http://www.who.int/alliance-hpsr/resources/publications/CLD_Course_Participant_Manual.pdf (дата обращения: 05.05.2018).