XV международная конференция
SECR / РАЗРАБОТКА ПО
14–15 ноября 2019 года, СПб

KiCad: от рисования к программированию

А. Н. Павлов

ФГУ ФНЦ НИИСИ РАН

Об авторе

Павлов А. Н. <antony@niisi.msk.ru>
сотрудник сектора Программирования НИИСИ РАН, участвую в верификации при помощи СПО (свободное программное обеспечение) разрабатываемых в НИИСИ микропроцессоров с MIPS-подобной системой команд (AKA архитектура КОМДИВ).

Antony Pavlov <antonynpavlov@gmail.com>
участник нескольких проектов СПО:

- linux;
- barebox (U-Boot v2);
- qemu;
- openocd;
- Zephyr-RTOS.

см. также
https://www.openhub.net/accounts/antonynpavlov

Участвуя в InnovateFPGA 2018, столкнулся с необходимостью быстро разработать простую печатную плату.
С тех пор сделал больше десятка плат при помощи KiCad, в процессе работы обнаружил, что приёмы, используемые при разработке ПО, могут быть перенесены в разработку аппаратного обеспечения.

- http://www.innovatefpga.com/cgi-bin/innovate/teams2018.pl?Id=EM099
- https://github.com/open-design/bus-spider-terasic-adapter

Постановка задачи

Средство разработки печатных плат на основе open-source компонентов (в первую очередь KiCad), которое обеспечивает пользователя поддержкой при синтезе принципиальной схемы.

Альтернативные САПР для проектирования электронных устройств

Существуют как открытые, так и коммерческие (весьма недешёвые) альтернативы KiCad

- gEDA/Lepton EDA

- Altium Designer®
- Autodesk® EAGLE™
- Cadence® Allegro®/OrCAD®
- DipTrace
- Mentor Graphics® PADS®/Xpedition®
- Proteus Design Suite

В дорогих платных САПР доступны элементы ИИ, в open-source нет.

Почему KiCad?

- KiCad de facto самая популярная open-source САПР для печатных плат.
  - ведётся активная разработка;
  - постоянно пополняется библиотека компонентов;
  - поддержка DigiKey;
  - отток пользователей EAGLE.
- в силу своей open-source природы KiCad отлично подходит для экспериментов.

Пример системы, спроектированной в KiCad

TERES-I Do-It-Yourself Free Open Source Hardware and Software laptop with ARM64 processor.

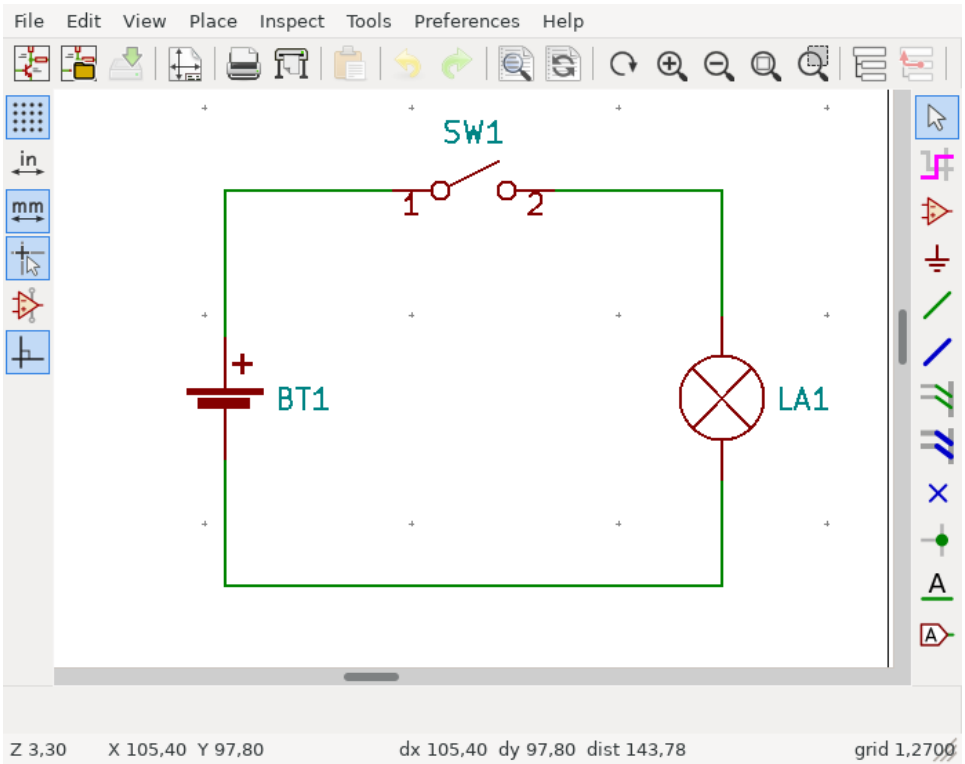
https://www.olimex.com/Products/DIY-Laptop/

материнская плата этого ноутбука разработана в KiCad:

TERES-PCB1-A64

KiCad: принципиальные схемы

- принципиальные схемы создаются в графическом редакторе eeschema;
- в eeschema отсутствуют возможности автоматизации.



Как это было 50 лет назад (1)

Печатный монтаж в радиолюбительских конструкциях

Инж. В. ФРОЛОВ

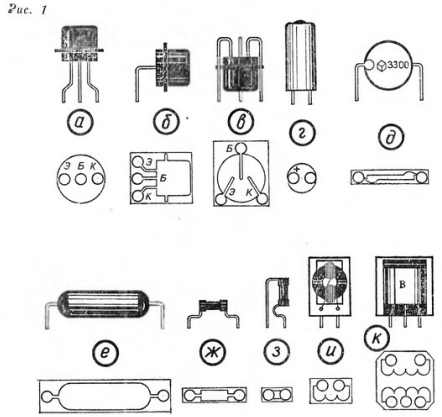
Применение печатных плат позволяет уменьшить габариты радиоустройства, свести к минимуму монтажные работы, в значительной степени избавляет от ошибок в монтаже, упрощает настройку и повышает надежность конструкции.

Для радиолюбителей наибольшую трудность при конструировании печатной платы представляет компоновка деталей — такое размещение их на плате, при котором наилучшим образом используется площадь платы, соединительные проводники проходят кратчайшим путем и не пересекаются. При этом должно быть учтено взаимное влияние элементов, чтобы готовое устройство не самовозбуждалось.

В любительских условиях наиболее доступным является аппликационный метод компоновки элементов схемы. Аппликации представляют собой вырезанные из плотной бумаги контурные рисунки деталей в натуральную величину в различных проекциях (рис. 1). Число аппликаций равно числу деталей устройства.

Для деталей простой формы (резисторы УЛМ-0,12, МЛТ-0,25÷0,5 и т. п.) и специально предназначенных для печатного монтажа (электролитические конденсаторы ТС 342, ТС 942 фирмы «Тесла», малогабаритные промышленные трансформаторы НЧ и ПЧ и т. п.) достаточно иметь аппликации одной проекции. Для деталей же, монтаж которых возможен в различных положениях (транзисторы, конденсаторы переменной емкости и т. п.), необходимо изготавливать аппликации, соответствующие проекциям деталей в этих положениях. Диаметр кружков-выводов деталей должен быть не менее 2,5 *мм*, что соответствует допустимому минимальному диаметру контактных площадок на проводниках печатной платы.

Рис. 1. Радиодетали, подготовленные к монтажу на печатной плате, и их аппликации: а, б, в — маломощные транзисторы; г — электролитический конденсатор фирмы «Тесла»; д — конденсатор типа КДС; е — конденсатор типа МБМ; ж, з — резисторы УЛМ — 0,12; и — катушка индуктивности на ферритовом кольце в арматуре; к — малогабаритный низкочастотный трансформатор.



Для облегчения компоновочных работ на принципиальной схеме надо пронумеровать точки соединения выводов элементов (рис. 2). Соответствующие номера вписываются и в кружки деталей на аппликациях. На аппликациях, кроме того, дополнительно можно указать позиционное обозначение деталей по схеме, цоколевку транзисторов, полярность электролитических конденсаторов.

Перед компоновкой необходимо определить максимально возможные размеры печатной платы. При этом, как и в случае с обычным монтажом (см. «Технологические советы», «Радио», 1966, № 9), намечаются точки крепления платы, вырезы под детали, устанавливаемые на стенках футляра. После этого, наметив форму платы и определив ее точные размеры, наносят контуры платы на плотную бумагу и приступают к компоновке.

При компоновке платы, размеры которой заданы, задача сводится к наиболее рациональному размещению аппликаций внутри ее контура. Если же основным требованием является возможно лучшее использование площади платы (конструирование малогабаритных устройств), то ее размеры можно определить только после окончания компоновки. В этом случае раскладку аппликаций надо производить на листе бумаги, предварительно начертив на нем две взаимно перпендикулярные линии (оси координат). После компоновки контур платы замкнуть и измерить размеры будущей платы.

Перемещая и поворачивая аппликации, надо всемерно стремиться к тому, чтобы контуры кружков (выводы деталей), обозначенные одинаковыми номерами, располагались возможно ближе друг к другу (рис. 3, *а*). Если выводы деталей не должны соединяться, а контуры их кружков окажутся на расстоянии менее 1 *мм*, тогда, уже на фольгированном материале, соответствующие контактные площадки придется подрезать, как показано на рис. 4.

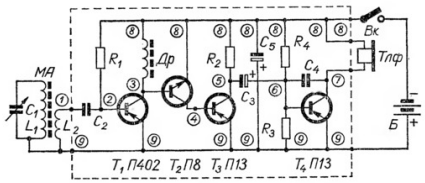


Рис. 2. Принципиальная схема транзисторного радиоприемника (В. Серов, «Простой карманный приемник», «Радио», 1964, № 8). Пунктиром обведена часть схемы, монтируемая на печатной плате.

Разместив аппликации, их приклеивают к бумаге резиновым клеем (удобен тем, что приклеенные им аппликации при необходимости легко отклеить). Затем на контур платы с аппликациями следует наложить лист кальки и перерисовать на него контуры платы, деталей и контактных площадок на их выводах. Перевернув кальку, с другой стороны проводят соединительные линии (рис. 3, б), стремясь при этом к тому, чтобы они не пересекались и были возможно короткими. Минимальную ширину линий токонесущих проводников и зазоры между соседними проводниками не следует делать менее 1 *мм*.

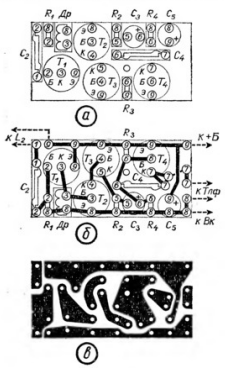


Рис. 3. Последовательность разработки печатной платы.

В местах вынужденных пересечений одну из соединительных линий можно разорвать и на ее концах начертить дополнительные контактные площадки (например 7—7 на рис.3,б). При монтаже они соединяются проволочной перемычкой со стороны размещения навесных деталей. Соединительным проводникам) удалять острием перочинного ножа или скальпеля, тонкие слои краски утолщать дополнительным нанесением краски.

Травление печатной платы производится раствором хлорного железа в воде плотностью 1,3 (в 200 *куб. см.* воды растворить 150 *г* хлорного железа в порошке). Готовый раствор выливают в плоскую стеклянную, пластмассовую или эмалированную посуду (например в кювету или ванночку для фоторабот) и в него погружают заготовку печатной платы.

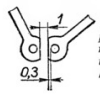
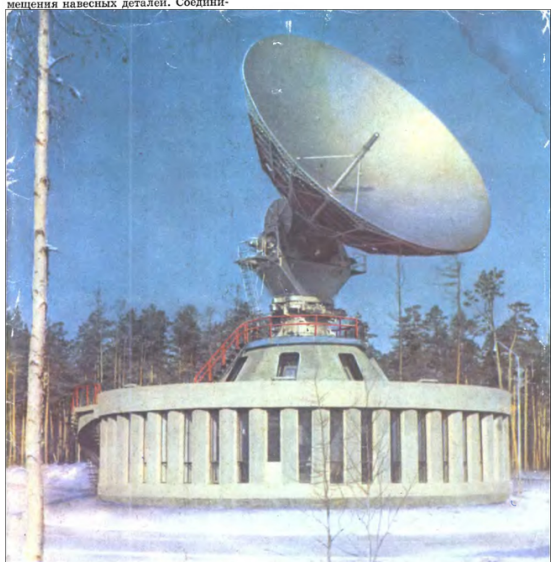


Рис. 4. Допустимые расстояния между контактными площадками печатных проводников.



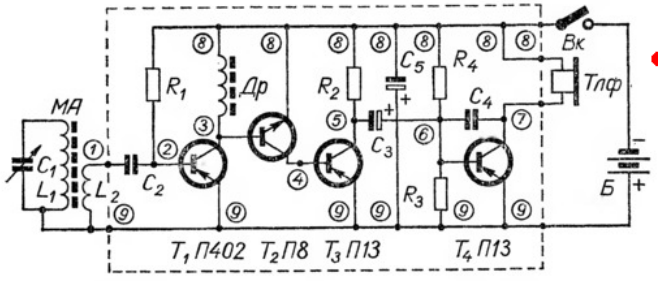
1 ЯНВАРЬ 1968
РАДИО

В НОМЕРЕ:

Защита Отечества — священный долг советских граждан ☆ Штурм космоса продолжается ☆ Горизонты науки. Живой мир подсказывает ☆ Передатчик начинающего ультракоротковолновика ☆ Блоки цветного телевизора ☆ Стандартизация в радиоэлектронике ☆ Любительский ламповo-полупроводниковый телевизор ☆ Для вас, автолюбители ☆ Цветомузыкальная установка ☆ Новый переносный радиоприемник «Меридиан»

Фролов В. Печатный монтаж в радиолюбительских конструкциях // Радио. 1968. №1. С. 42-43.
http://archive.radio.ru/web/1968/01/042/

Как это было 50 лет назад (2)



Принципиальная схема

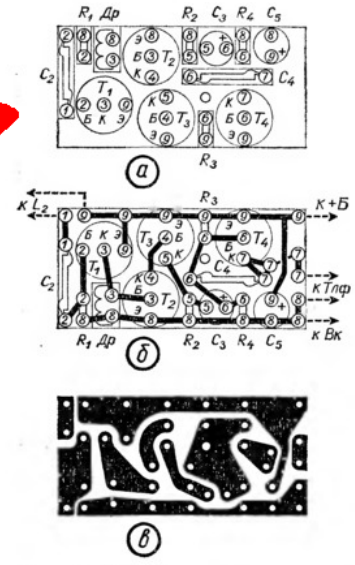
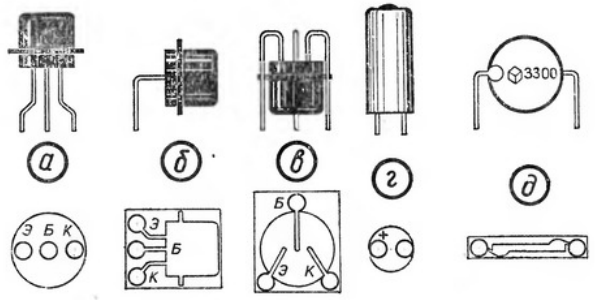
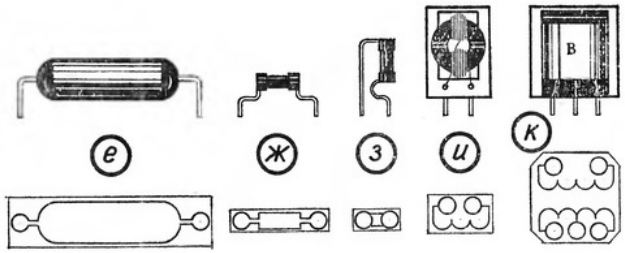


Рис. 3. Последовательность разработки печатной платы.





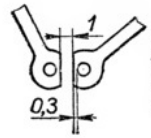
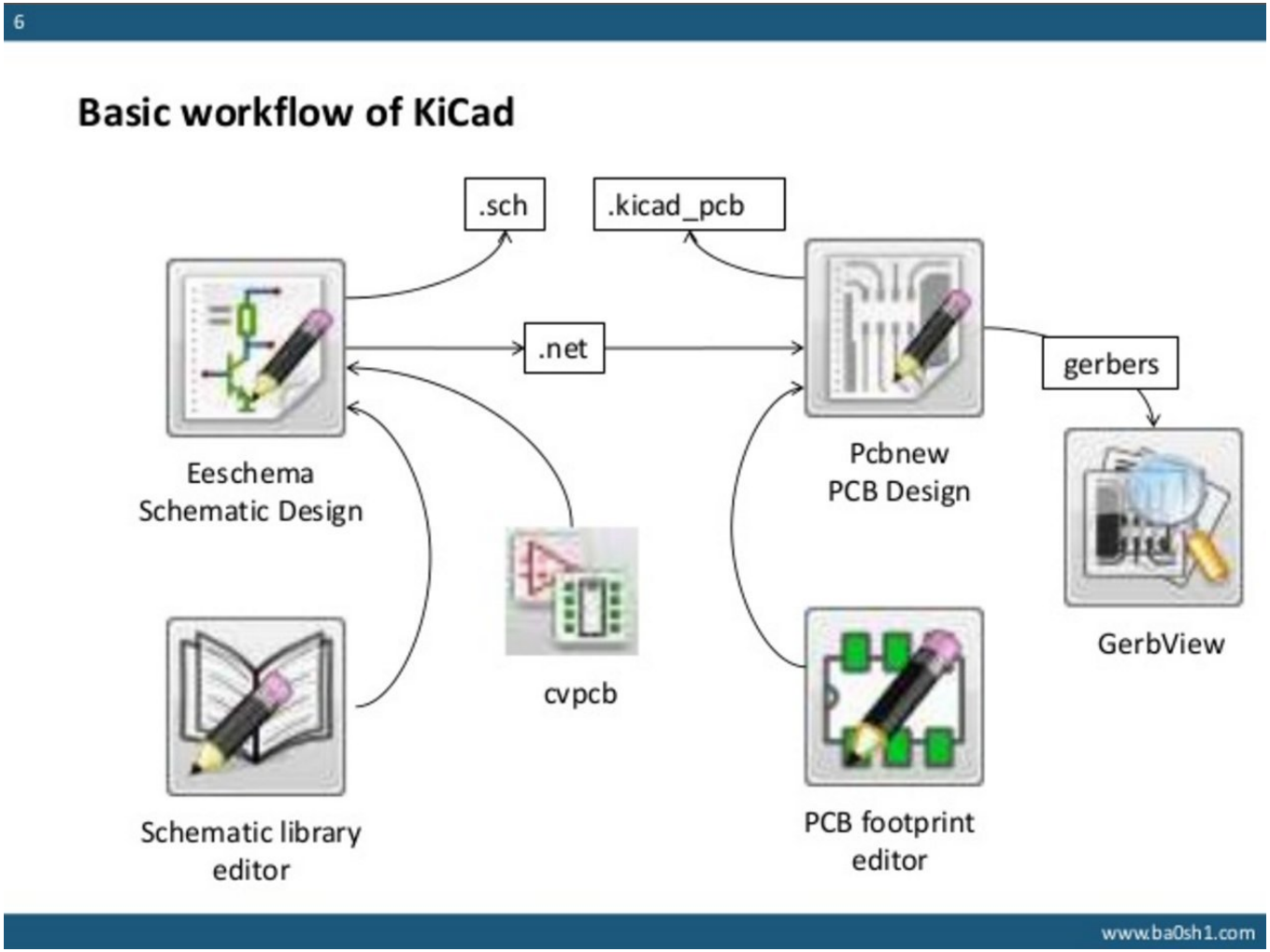


Рис. 4. Допустимые расстояния между контактными площадками печатных проводников.

DRC

DRC — Design Rule Checking

Что изменилось за 50 лет?



ОТВЕТ: ПОЧТИ НИЧЕГО!

Что изменилось за 50 лет? (2)

1) разработчик по прежнему рисует схемы вручную, только с использованием технических средств;

2) отсутствуют технологии ИИ в open-source САПР для разработки принципиальных схем;

3) как итог, низкая производительность труда.

Примечание: средства прокладывания трасс всё-таки появились.

Предлагается решение на основе open-source по оснащению САПР KiCad подсистемой с элементами ИИ для интеллектуализации работы разработчика электроники.

Декларативный и императивный подходы

декларативный подход

Программист определяет, что будет вычислено.

императивный подход

Программист определяет, как будет вычислено то, что надо вычислить.

Декларативный подход

```
var numbers = [1,2,3,4,5,6,7,8,9,0];

var odds = _.filter(numbers, (n) =>
{ return n % 2 === 0; });
```

Императивный подход

```
var numbers = [1,2,3,4,5,6,7,8,9,0];

var odds = [];

numbers.forEach(function(n) {
    if (n % 2 === 0) {
        odds.push(n);
    }
});
```

Важная роль логики в ИИ заключается в том, что она представляет инфраструктуру для решения задачи в декларативном стиле. Так, человек-эксперт подробно излагает своё знание предметной области при помощи описательной логики и использует системы логического вывода для решения вычислительных задач в этой области.

Об индукции, дедукции и абдукции(2)

Дедукция

Способ рассуждения, при котором новое положение выводится логическим путём от общих положений к частным выводам. Переход от общих положений, законов и т. п. к частному, конкретному случаю

Индукция

Переход от частного к общему, от единичного наблюдения к обобщению, когда общее заключение выводится на основе множества посылок, сформированных по накопленному опыту.

Абдукция

Процесс логического вывода, результатом которого является формирование гипотез, объясняющих наблюдаемые феномены.

Примеры дедукции, индукции и абдукции

дедукция

Правило: Если пациент принимает лекарство, то он выздоравливает.
Известный факт: Этот пациент принимает лекарство.
Результат: Этот пациент выздоравливает.

индукция

Известный факт: Этот пациент принимает лекарство.
Результат: Этот пациент выздоравливает.
Правило: Если пациент принимает лекарство, то он выздоравливает.

абдукция

Правило: Если пациент принимает лекарство, то он выздоравливает.
Результат: Этот пациент выздоравливает.
Гипотеза: Наверное этот пациент принимает лекарство.

Абдукция в реальной жизни

Два основных применения абдукции: диагностика и синтез в условиях неполной информации.

Абдукция нашла широкое применение в ИИ при решении задач диагностики, понимании естественного языка, планирования, пополнения и пересмотра знаний в базах знаний, в мультиагентных системах.

Пролог — язык логического программирования

Prolog отличается от привычных языков вроде C, C++, Java:

- основан на логике
  - программа на Прологе описывает не процедуру решения задачи, а логическую модель предметной области;
  - программы на Пролог как правило короткие;
- символьные вычисления (а не числовые!);
- широко используется в системах ИИ (например, бронирование авиабилетов).

Про Пролог и CHR

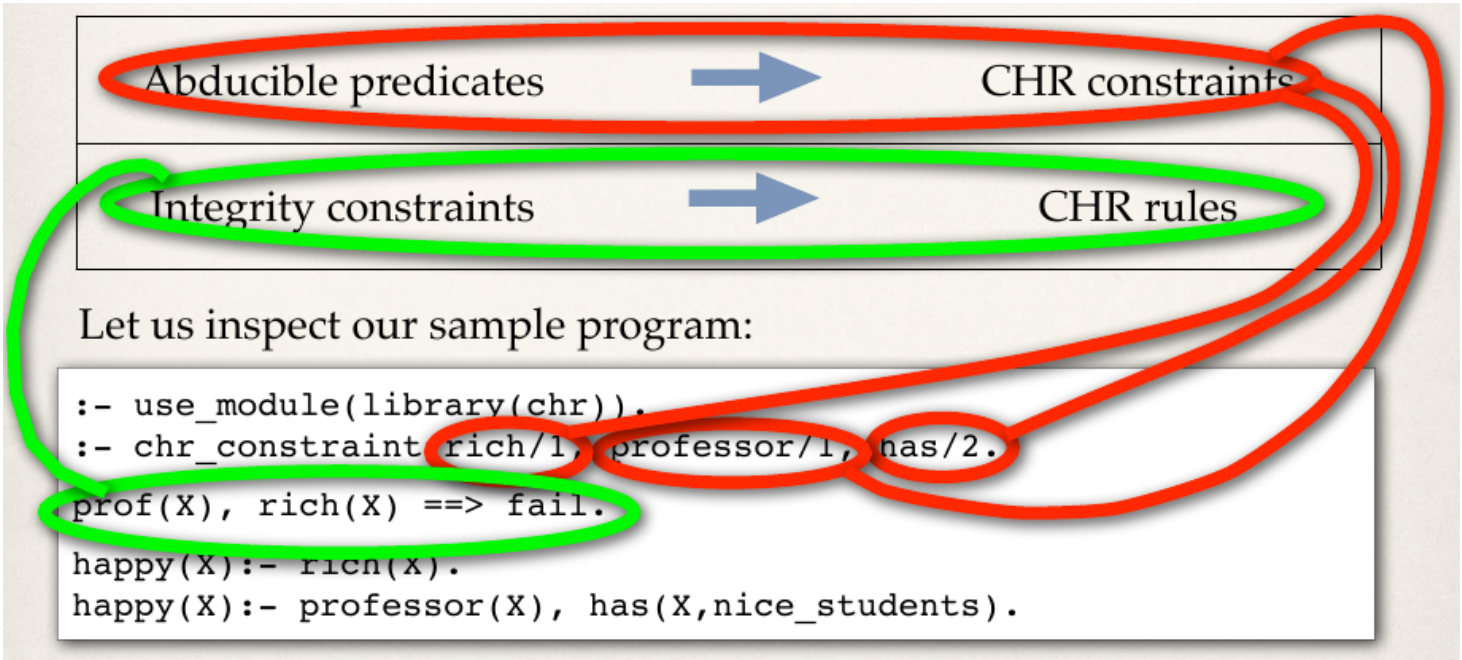
CHR — непосредственно исполняемые декларативные спецификации, библиотека для языка Пролог, позволяющая реализовывать решатели в ограничениях.
CHR позволяет реализовать абдукцию на Прологе, перейти к ALP (Abduction Logic Programming).
CHR вводит три типа правил:

```
Propagate       c, c, ... c => Guard | ... c ...
    добавление ограничений

Simplify        c, c, ... c <=> Guard | ... c ...
    замена ограничений

Simpagate       c, ... \ c, ... <=> Guard | ... c ...
    сокращение ограничений
```

Пример реализации абдукции в CHR (1)



Abduction and language processing with CHR (CHR Summer School – September 2010), Henning Christiansen
https://dtai.cs.kuleuven.be/CHR/summerschool/slides/christiansen.pdf

Пример реализации абдукции в CHR (2)

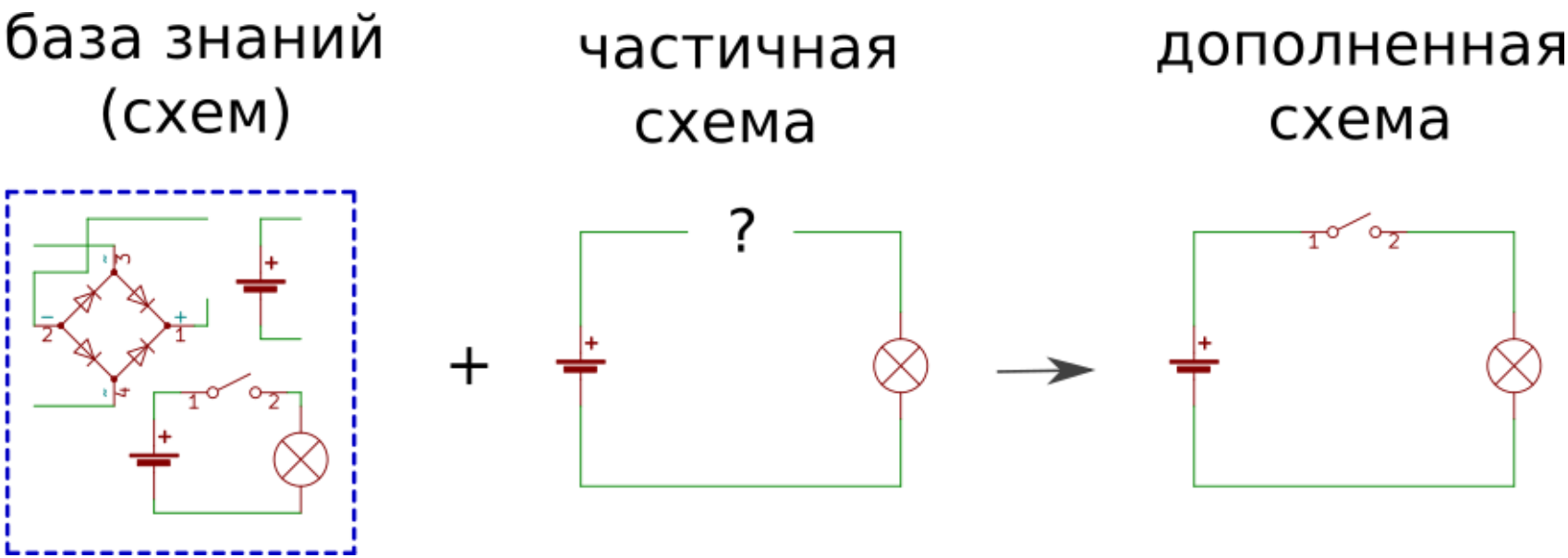
```
:- use_module(library(chr)).
:- chr_constraint rich/1, professor/1, has/2.

professor(X), rich(X) ==> fail.
happy(X) :- rich(X).
happy(X) :- professor(X), has(X, nice_students).

/*
 * ?- consult(secr_happy_professor).
 * true.
 *
 * ?- happy(henning), professor(henning).
 * professor(henning),
 * professor(henning),
 * has(henning, nice_students).
 *
 */
```

База знаний и абдуктивный вывод применительно к схемам

Пример абдуктивного вывода для схем (синтез по частичной спецификации, представленной ограничениями):



Самый короткий солвер, иллюстрирующий описанный подход

```
:- [library(chr)].
:- chr_constraint (@)/1.

%%%% база знаний
c(flashlight) :- @batt, @switch, @lamp.
c(radio) :- @batt, @radio.

@match, @batt ==> @c(flashlight) ; @c(radio).

@match, @switch ==> @c(flashlight).
@match, @lamp ==> @c(flashlight).
@match, @radio ==> @c(radio).
%%%%

@c(_) \ @match <=> true.
@X \ @X <=> true.

recognize(ComponentsGiven, CircuitCandidate) :-
 @match,
 call(ComponentsGiven),
 find_chr_constraint(@c(CircuitCandidate)).

abduce_circuit(ComponentsGiven, AbducedCircuit) :-
 recognize(ComponentsGiven, Circuit),
 c(Circuit),
 findall(C, find_chr_constraint(C), AbducedCircuit).
```

Самый короткий солвер: запросы (1)

Примеры запросов (указываем какие-то компоненты, и получаем абдукцией схемы-кандидаты):
1) батарея подходит для двух схем:

```
?- abduce_circuit((@batt), Circuit).
Circuit = [@lamp, @switch, @c(flashlight), @batt],
@lamp,
@switch,
@c(flashlight),
@batt ;
Circuit = [@radio, @c(radio), @batt],
@radio,
@c(radio),
@batt ;
false.
```

Самый короткий солвер: запросы (2)

2) радио подходит только для одной схемы:

```
?- abduce_circuit((@radio), Circuit).
Circuit = [@batt, @c(radio), @radio],
@batt,
@c(radio),
@radio ;
false.
```

3) незнакомый компонент не подходит ни одной схеме:

```
?- abduce_circuit((@mcu), Circuit).
false.
```

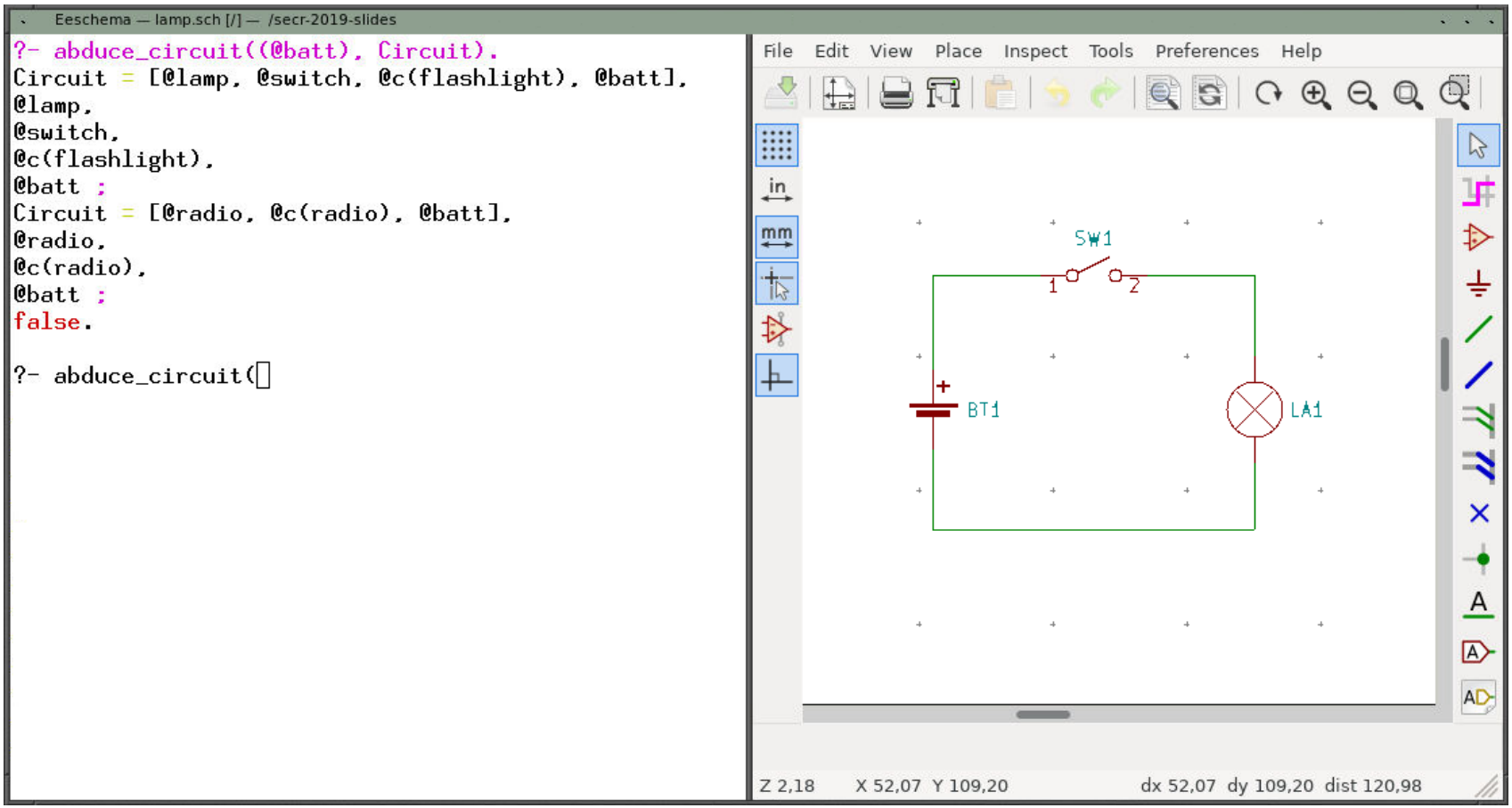
Самый короткий солвер: запросы (3)

4) можно указать два компонента, один из которых дополнится до известной схемы:

```
?- abduce_circuit((@mcu, @radio), Circuit).
Circuit = [@batt, @c(radio), @radio, @mcu],
@batt,
@c(radio),
@radio,
@mcu ;
false.
```

Мультимодальный интерфейс пользователя

Интерфейс пользователя: мультимодальный ввод информации (ком строка и графический редактор схем)



Новый WORKFLOW

- добавление компонентов с помощью мультимодального UI
- интеграция данных об указанных компонентах в систему абдуктивного вывода
- идентификация схем-кандидатов из базы знаний, подходящих под паттерны, указанные с помощью UI
- устранение схем-кандидатов, нарушающих ограничения целостности
- дополнение добавленных компонентов компонентами из базы знаний о схемах
- поиск подходящих изображений компонентов в библиотеке
- размещение изображений компонентов на листе
- генерация файла принципиальной схемы eeschema

Заключение

- преимущества по сравнению с простым редактором схем
- преимущества по сравнению с другими подходами:
  - skidl (Python) [skidl] [skidltalk];
  - jitX (jitx.com).
- получение частичных решений, online-алгоритм, anytime-алгоритм.

- Павлов А. Н. <antony@niisi.msk.ru>
- Antony Pavlov <antonynpavlov@gmail.com>
- https://github.com/frantony
- https://www.openhub.net/accounts/antonynpavlov