

4. Солодкий, Д.М. Эффективное покрытие мобильных платформ с использованием технологии Apache Cordova / Д.М. Солодкий, В.Н. Козуб. – Материалы 51-я научной конференции. – Минск: БГУИР, 2015. – С. 35.
5. Солодкий, Д.М. Программный модуль агрегации расписаний с использованием нереляционной базы данных: дипломная работа / Д.М. Солодкий. – Минск: БГУИР, 2015. – 100 л.
6. About HTML5 WebSocket [Электронный ресурс]. – Режим доступа: <http://www.websocket.org/aboutwebsocket.html>
7. Nodemon [Электронный ресурс]. – Режим доступа: <http://nodemon.io/>.

ДЕКЛАРАТИВНОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PROLOG

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Чистяков Михаил

Луцик Ю. А. – к-т технич. наук, доцент

Декларативное программирование сильно отличается от того подхода, к которому привыкли многие программисты. Практически все популярные на сегодняшний день языки программирования реализуют **императивный** подход. Суть этого подхода заключается в том, что программист должен разработать алгоритм решения задачи и правильно оформить его в рамках языка. Однако само построение алгоритма уже является достаточно непростой творческой задачей. Парадигма **декларативного** программирования даёт возможность абстрагироваться от построения алгоритма и сосредоточиться на правильном формальном описании задачи. Рассмотрим основные моменты данного подхода на примере языка **Prolog**.

Программа на языке Prolog состоит из фактов, правил и вопросов. Правило представляет собой конструкцию вида “заголовок :- тело” и буквально означает “заголовок истина, если тело истина”. Факт представляет собой предикат с конкретным значением и имеет вид “имя(значение)”. Совокупность фактов и правил программы составляют её **базу знаний**. Она имеет ключевое значение в процессе решения задачи. Программа, представляющая собой базу знаний, может иметь много точек входа. Решение инициируется при получении некоторого запроса, по которому производится поиск в базе знаний. В основе поиска решения программы лежит метод **поиска с возвратом** и **унификация**. Если есть несколько вариантов дальнейшего выполнения (определено несколько правил с одинаковой сигнатурой), то будут по очереди проработаны все варианты. Если при выполнении программы будет получено противоречие, либо требуемый факт вернёт значение ‘ложь’, произойдёт возврат назад в то место, в котором возможно было другое поведение. Рассмотрим применение Prolog для решения следующей задачи.

В некотором поле шахматной доски 8x8 находится шахматный конь. Найти последовательность ходов, которая может привести его в другое указанное поле.

Ниже приведён один из возможных вариантов реализации данной программы на языке SWI-Prolog. Для того, чтобы получить желаемый ответ, необходимо сформировать запрос вида “findSolutions([x1,y1], [x2,y2], S, X).” Ответ будет представлен в виде X=[[x',y'],[x'',y''],...end], где [x',y'] – координаты следующего хода.

1. findSolutions(StartPos, Goal, MinSteps,X):-
2. solve(StartPos, Goal, MinSteps,X);
3. NewSteps is MinSteps +1,
4. findSolutions(StartPos, Goal, NewSteps,X).
5. solve(X,X,0,[end]).
6. solve(StartPos,Goal,StepsLost,[NewPos|Tail]):-
7. go(StartPos,NewPos,8),
8. NewSteps is StepsLost-1,
9. NewSteps >=0,
10. solve(NewPos,Goal,NewSteps ,Tail).
11. go([X,Y],[X1,Y1],M):- X1 is X+1, Y1 is Y+2, X1<M, Y1<M.
12. go([X,Y],[X1,Y1],M):- X1 is X+2, Y1 is Y+1, X1<M, Y1<M.
13. go([X,Y],[X1,Y1],M):- X1 is X+2, Y1 is Y-1, X1<M, Y1>=0.
14. go([X,Y],[X1,Y1],M):- X1 is X+1, Y1 is Y-2, X1<M, Y1>=0.
15. go([X,Y],[X1,Y1],M):- X1 is X-1, Y1 is Y-2, X1>=0, Y1>=0.
16. go([X,Y],[X1,Y1],M):- X1 is X-2, Y1 is Y-1, X1>=0, Y1>=0.
17. go([X,Y],[X1,Y1],M):- X1 is X-2, Y1 is Y+1, X1>=0, Y1<M.
18. go([X,Y],[X1,Y1],M):- X1 is X-1, Y1 is Y+2, X1>=0, Y1<M.

Данный декларативный подход отлично работает для задач, алгоритм решения которых не определён. Prolog успешно справляется с задачами по прокладыванию маршрута или поиску необходимой расстановки. Представление программы в виде базы знаний и набора запросов делает Prolog мощным инструментом для работы с различными базами данных, а также для работы с искусственным интеллектом. Таким образом, знакомство с Prolog-ом будет полезно для программистов любой квалификации для расширения арсенала методов решения задач и взглядов на программирование в целом.

Список использованных источников:

1. Сошников Д.В. Парадигма логического программирования / В.Д. Сошников // М.: "Вузовская книга", 2006. – 220 с.
2. Братко И. Программирование на языке Пролог для искусственного интеллекта / И. Братко // М.: "Мир", 1990 – 560 с.

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС ИМИТАЦИИ ПРОЦЕССА ФУНКЦИОНИРОВАНИЯ РЛС ЗРК В УСЛОВИЯХ ВОЗДЕЙСТВИЯ АКТИВНЫХ ПОМЕХ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь

Лапука А. О.

Лукашевич М. М. – к. т. н.

Одной из бурно развивающихся в настоящее время областей реализации научно-технического потенциала в оборонном секторе отечественной экономики является развитие средств индивидуальной радиоэлектронной защиты летательных аппаратов от управляемого ракетного оружия (УРО) зенитных ракетных комплексов (ЗРК) и авиационных ракетных комплексов перехвата (АРКП). Проведение натурных испытаний таких систем сопряжено с достаточно серьезными организационными трудностями и существенными финансовыми затратами: требуется подготовка и выполнение исследовательских (испытательных) полетов, организация системы управления экспериментом, сбора объективной информации и т.д. Кроме того, в результате летных испытаний обычно не удается получить требуемый объем информации, характеризующий эффективность функционирования разрабатываемой аппаратуры в широком диапазоне условий ее боевого применения.

Для преодоления перечисленных проблем разработчиками используется подход, основанный на методе полунатурных испытаний. Он характеризуется разработкой программно-аппаратного комплекса сигнальной имитации внешних (по отношению к разрабатываемому оборудованию) систем, обеспечивающих воспроизведение условий его боевого применения. Основными требованиями, предъявляемыми к такому комплексу, являются:

- разработка и программная реализация сценариев имитации целевой и помеховой обстановки, учитывающих взаимное расположение и перемещение подавляемой РЛС и помехопостановщика, динамику изменения ракурса наблюдения,

- программно-управляемая имитация зондирующего сигнала подавляемой РЛС с наделением его необходимым видом модуляции, временными и спектральными характеристиками. При этом принципиально необходимыми условиями являются наличие излучения сформированного сигнала на литерной (несущей) частоте, с учетом временной задержки и масштабирования по мощности в соответствии с текущей дальностью расположения РЛС относительно помехопостановщика, а также с учетом угла поворота приемно-передающей антенны (Рисунок 1). Это необходимо для обеспечения корректности имитации сигналов, поступающих на вход приемника разрабатываемой аппаратуры;

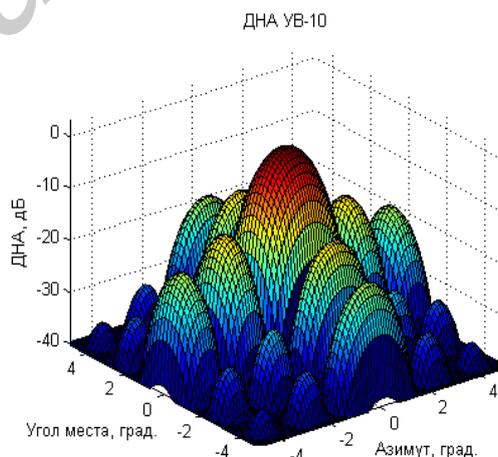


Рисунок 1 – диаграмма направленности приемно-передающей антенны УВ-10

- наличие имитатора полезного сигнала, обеспечивающего соответствующий заданной целевой обстановке уровень мощности и спектрально-временные характеристики отраженных сигналов от цели;
- наличие устройства приема и частотного преобразования полезных и помеховых сигналов,