International Journal of Mechatronics, Electrical and Computer Technology

Vol. 2(5), Oct, 2012, pp 111-121, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

An Expert System Approach for Garden Designing

Niloofar Mozafari*, Zeinab Khorshidpour, Shahram Jafari

Department of Computer Science, Engineering and IT, School of Electrical Computer Engineering, Shiraz University, Shiraz, Iran

*Corresponding Author's Email: *Mozafari@cse.shirazu.ac.ir*

Abstract

In the recent years, the quality of human life is improved by artificial intelligence techniques. In artificial intelligence, an expert system is a computer system that emulates the decision-making ability of a human expert.Expert systems are designed to solve complex problems by reasoning about knowledge, like an expert. In this paper, we propose an expert system with the aim of designing the garden with considering the different taste of the people. The proposed system can help people to design their garden themselves. Indeed, it is able to use by architectures to provide decision support system, interactive training tool and expert advice. The system constitutes part of intelligent system of designing the garden. An initial evaluation of the expert system was carried out and a positive feedback was received from the users.

Keywords: Artificial Intelligence, Expert Systems, CLIPS, Garden Design.

1. Introduction

Computer-based methods are increasingly used to improve the quality of human life. One of the most important areas of computer science that focuses on creating machines that can mimic the behavior of human is artificial intelligence [1]. The ability to create intelligent machines has intrigued humans since ancient times and today with the advent of the computer and 50 years of research into AI programming techniques, the dream of smart

Vol. 2(5), Oct, 2012, pp 111-121, ISSN: 2305-0543
Available online at: http://www.aeuso.org

machines is becoming a reality.Researchers are creating systems which can mimic human thought, understand speech, beat the best human chess player and countless other feats never before possible [2]. Dependence on the human expert can be minimized if his/her expertise can be transferred into a computer system. The proposed system for designing a garden is an expert system. An expert system is a system that employs human knowledge captured in a computer to solve problems that ordinarily require human expertise [3]. Expert system seeks and utilizes relevant information from their human users and from available knowledge bases in order to make recommendations [4].With the expert system; the user can interact with acomputer to solve a certain problem. This canoccur because the expert system can store heuristicknowledge [5].

The development of expert system is implementedin CLIPS programming environment [6]. CLIPS is the abbreviation of C LanguageIntegrated Production System [7]. Thisprogramming tool is designed to facilitate thedevelopment of software to model humanknowledge or expertise for medical therapy.CLIPS program is used by reason of the flexibility,the expandability and the low cost.There are many types of garden styles, and opinions on what makes a garden aesthetically appealing, lot of different uses for gardens, and all kinds of gardeners with different experiences who are working with different kinds of environments. All these variables donot even begin to cover the complexity in working with a palette of plants that easily numbers in the thousands (if not tens of thousands) just for common specimens. The area that we choose to focus on was that of a garden bed and its composition. Our new expert system isnot strictly a planner; it won't come up with a garden bed by scratch. Instead a user enters plants that they have in an existing bed, and the system will return a list of plants that will help improve the composition of the bed. Composition is defined here as the structure of the bed, where every plant serves a certain purpose, and the bed has the right proportion of plants serving that purpose, achieving a balance. There are other factors besides composition that help create a good bed.

Vol. 2(5), Oct, 2012, pp 111-121, ISSN: 2305-0543
Available online at: http://www.aeuso.org

According to our knowledge no one designed expert systems for designing the garden, so in this work, we will present a design of an expert system for designing garden using CLIPS. We preset in Section 2 the preliminaries of our method.Section 3 explained our expert system for garden designing. The result of our method is proposed in Section 4 and paper concludes in Section 5.

2. Preliminaries

Many different purposes can be served bygardens from a casual kitchen herb garden, based upon functionality and casualness, to historical formal knot gardens, designed to keep hoards of common people busy and to impress other nobles with their grandeur.Each function and style has its own criteria for what is considered beautiful. Recently people are interested in two types of gardens: the purely functional vegetable gardens or easy to care for yet beautiful landscapes. It should be mentioned that there is no need to have an expert to plant a vegetable garden. The user just plants what she/he wants to eat of vegetables that can be grown in the area. But there are too many choices and options when designing landscapes.

One type of garden design is to use mixed borders. These borders are composed of many different types of plants with the idea being that it is easier to care for many different species than a homogeneous bed. In this case, it is good to benefit from an expert system that helped people to create this type of bed. There are many guidelines to keep in mind when designing a good border. For example, a mixed border is composed of several different types of plants which each one serves a specific goal. In the first step, the design starts with plants that are medium height which acts as a focal point. Then some evergreen shrubs can add to accent it. Both of these types of plants are made the structure of the garden. In order to add color and variety, some perennials can be added to the garden. They planted once and will come back year after year, but typically die back to the ground in winter, and donot get very big. Groundcovers act to tie everything together like a carpet and help keep out weeds.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 2(5), Oct, 2012, pp 111-121, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

Vines are an easy way to add height to a garden to keep it from looking too horizontal. Bulbs are another glamorous plant. They tend to send out spectacular flowers but die back after that. It should be mentioned that most people love annuals. They are plants that live for only one year; however, they make up for their short lives by blooming profusely. These plants createa lot of color, but they tend to require attention in the form of constant deadheading and planting and digging up every year.A mixed border should have most of these things, and it should have them in a good balance. There exist also many different features that can help to expert for designing a good garden,for example seasonal interest, texture, color, environment and etc. Our expert system takes into account all of them and suggest a good design for garden.

3. Proposed method

Our expert system for designing garden uses knowledge base to recommend a design for garden. Figure 1 illustrates the structure of expert system that is developed in this work.

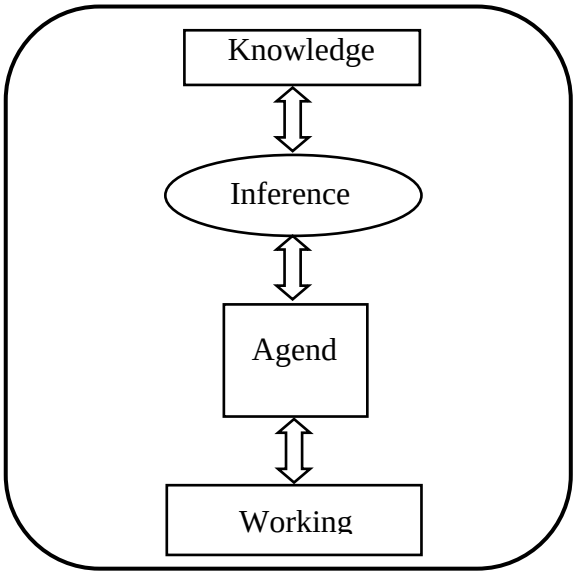

Figure 1: the structure of our Expert System

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 2(5), Oct, 2012, pp 111-121, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

We implement the problem of garden designing using the methodology of rule based systems. One of the well-known methods of representation of knowledge in the expert systems is the productive representation as the CLIPS [6]. CLIPS is a public domain software tool for building expert system.It is probably the most widely used expert system tools because itis fast, efficient and free. CLIPS incorporates a complete object-oriented languagefor writing expert systems. Like other expert system languages, CLIPSdeals with rules and facts. Various facts can make a rule applicable. An applicable ruleis then asserted. Facts and rules are created by first defining them.CLIPS keeps in memory a fact list, a rule list, andan agenda with activations of rules.

The facts inCLIPS are simple expressions consisting of fieldsin parentheses. Groups of facts in CLIPS, usuallyfollow a fact-template, so that to be easy toorganize them and thus design simple rules thatapply to them.The algorithm of our expert system is straight forward. It comes with 60 plants built in. Knowledge about the plants fits into a predefined template. Our expert system contains 30CLIPS rules. Below, we present some of the rules for designing the garden.

```
;;…….…… The initial Rule ……..……..……
(defrule start
  ?init <- (init-fact)
  =>
  (printout t "Welcome to Garden Design System :) !" crlf
          "The available plants are listed as bellow : " crlf)
  (retract ?init)
  (assert (print-ava-list list))
  )
;;…….…… List of Available Plants ……..……
(defrule print-ava-list
  (print-ava-list list)
  (plant (name ?name))
```

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 2(5), Oct, 2012, pp 111-121, ISSN: 2305-0543
Available online at: http://www.aeuso.org


```
 =>
 (printout t "     " ?name crlf)
 )
;;;……..…… Asking the User for Ready to Design ……...……..……..……
(defrule ready-to-design
  ?print <- (print-ava-list list)
=>
  (retract ?print)
  (printout t "Ready to design : (yes/no) ?" crlf)
  (assert (ready-to-design =(read)))
  )
;;;……..…… Getting List of Existing Plants ……..……
(defrule user-is-ready
 ?ready-to-design <- (ready-to-design yes)
 =>
 (retract ?ready-to-design)
 (printout t "In order to determine what new plants to add to an existing bed," crlf
                                  "please enter all of the plants that are located there:" crlf)
 (bind ?string (readline))
 (assert-string (str-cat "(existing-plants " ?string ")"))
 (assert (num-plants (length$ (explode$ ?string))))
 )
;;;……..…… Computing Border Usage ……..……..……..……
(defrule compute-usuage
        (existing-plants $?first ?aPlant $?end)
        (plant (name ?aPlant) (usage ?aUse))
        (not (usage-fired ?aPlant))
        ?counter-rule <- (counter ?aUse ?counter)
        ?total-rule <- (counter total-plants ?tot-plants)
        =>
        (assert (usage-fired ?aPlant))
```

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 2(5), Oct, 2012, pp 111-121, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

- -

```
        (retract ?counter-rule)
        (retract ?total-rule)
        (assert (counter total-plants (+ ?tot-plants 1)))
        (assert (counter ?aUse (+ ?counter 1))))

;;…… Checking Bed to need more Backbone Plant ……...……
(defrule compute-backbone
        (counter backbone ?num-backbone)
        (compute-usage done)
        (counter total-plants ?tot-plants)
        (test (> 0.3 (/ ?num-backbone ?tot-plants)))
        =>
        (assert (need backbone)))

;;…… Checking Bed to need more perennials ……...……
(defrule compute-perennial
        (compute-usage done)
        (counter perennial ?num-perennial)
        (counter total-plants ?tot-plants)
        (test (> 0.4 (/ ?num-perennial ?tot-plants)))
        =>
        (assert (need perennial)))

;;…….…… Checking Bed to need more annuals ……..……
(defrule compute-annual
        (compute-usage done)
        (counter annual ?num-annual)
        (counter total-plants ?tot-plants)
        (test (> 0.1 (/ ?num-annual ?tot-plants)))
        =>
        (assert (need annual)))

;;……..…….. Checking Bed to need more vines…….…….……
(defrule compute-vines
        (compute-usage done)
```

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 2(5), Oct, 2012, pp 111-121, ISSN: 2305-0543
Available online at: http://www.aeuso.org


```
        (counter vine ?num-vines)
        (counter total-plants ?tot-plants)
        (test (> 0.1 (/ ?num-vines ?tot-plants)))
        =>
        (assert (need vine)))

;;…….…… Checking Bed to need more bulbs plants……..……
(defrule compute-bulbs
        (compute-usage done)
        (counter bulbs ?num-bulbs)
        (counter total-plants ?tot-plants)
        (test (> 0.1 (/ ?numb-bulbs ?tot-plants)))
        =>
        (assert (need bulbs)))

;;…….……..… Checking Bed to need any groundcovers…….……
(defrule compute-groundcovers
        (compute-usage done)
        (counter groundcover ?num-groundcovers)
        (test (> 1 ?num-groundcovers))
        =>
        (assert (need groundcover)))
```

The program starts by displaying a list of plants that the system has knowledge of it. It then prompts the user for a list of plants that they already have in a garden bed. In order to make sure that the entered plants by user are ones that system has knowledge about them, an error-checking is done. Then, it analyzes the bed by looking at each individual plant and that plants usage. System counts up the number of backbone plants, perennials, annuals, vines, etc. It then determines what percentage each type makes up. The obtained values are compared to the ideal value and if it falls below the ideal a new rule is asserted. Then the system investigates different factors for garden designing and suggests the extra plants according to them.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 2(5), Oct, 2012, pp 111-121, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

Results

We create a user interface for communicating between the user and the system. This user interface was implemented in English language. The user interface is represented as a welcome message to the user and the list of all available plants. Figure 2 illustrates the startup menu of our system.

```
CLIPS> (reset)
CLIPS> (run)

Welcome to the Garden design system :)

The available plaints are listed as bellow:
 hyacinths
 snowdrops
 calla
 crocus
 daylily
 daffodil
 glads
 lily
 kiwi
 sweet-peas
 honeysuckle
 climbing-hydrangea
 wisteria
 morning-glories
 climbing-rose
 clematis
 sweet-alyssum
 cleome
 mexican-sunflowers
 snapdragons
 calendulas
 hollyhocks
```

Figure 2: the startup menu of our system.

International Journal of Mechatronics, Electrical and Computer Technology

Vol. 2(5), Oct, 2012, pp 111-121, ISSN: 2305-0543
Available online at: http://www.aeuso.org

When the user is ready to design, the program shows the list of available plants. In order to determine what new plants are needed to add the bed, the user must enter the existing plant in her/his bed. After that, system recommends a list of plants. Figure 3 displays a sample of system result.

```
    Are you Ready to design : (yes/no) ?
yes
In order to see our recommendation for adding the new plants to an existing bed,
please enter all of the plants that are located in your bed:
holly

We suggest adding the following plants to your garden.
(lamium kiwi sweet-alyssum blue-oat-grass)
CLIPS>
```

Figure 3: A sample of our system result.

In order to evaluation of our system, we compare the result of our expert system with an expert. The obtained results evaluates with the opinion of an architect and a plant expert. Both of experts were agree with the output and the obtained result was promising according to their view.

Conclusion

Expert system has many applications in different domain. In this paper, we propose an expert system for designing garden. The proposed system can help people to design their garden themselves. Indeed, it is able to use by architectures to providedecision support system, interactive training tool and expert advice. The system constitutes part of intelligent system ofdesigning the garden. An initial evaluation of the expert system was done by architectures and plant experts. A number of them tested the system and gave us a positive feedback and asked us to expand the expert system to cover more designing systems. As future work we will constitute the expert system to design house according to different interested of users.

Vol. 2(5), Oct, 2012, pp 111-121, **ISSN**: 2305-0543
Available online at: http://www.aeuso.org

References

[1] G. Luger,W. Stubblefield, "Artificial Intelligence: Structures and Strategies for Complex Problem Solving", the Benjamin/Cummings Publishing Company, Inc. ISBN 0-8053-4780-1, 2004.

[2] S. Russell, P.Norvig, "Artificial Intelligence: A Modern Approach", Upper Saddle River, New Jersey: Prentice Hall, ISBN 0-13-790395-2, 2003.

[3] P. Jackson, "Introduction to Expert Systems", Harlow, England: Addison Wesley Longman., third edition, 1999.

[4]E. Turban,"Expert System and Applied Artificial Intelligence", Macmillan Publishing Company, New York, 1992.

[5] S.L. Kendal, M. Creen, "An introduction to knowledge engineering", London: Springer, ISBN 978-1-84628-475-5, OCLC 70987401, 2007.

[6]J. Giarranto,"CLIPS User's Guide", Version 6.22, 1998.

[7] C. Giarratano, D. Riley, "Expert Systems: Principles and Programming", Fourth Edition, 2002.