Good CARMA for the High Plains

Alexandre V. Latchininsky
Dept. of Renewable Resources
University of Wyoming
Laramie, WY, USA
latchini@uwyo.edu

John D. Hastings
Dept. of Computer Science and Information Systems
Univ. of Nebraska-Kearney
Kearney, NE, USA
hastingsjd@unk.edu

Scott S. Schell
Dept. of Renewable Resources
University of Wyoming
Laramie, WY, USA
sschell@uwyo.edu

Abstract

CARMA is a decision-support system for grasshopper infestations that has been successfully used since 1996. Rising treatment costs coupled with shrinking rangeland profit margins increasingly demand accurate selection of the most cost-effective responses to grasshopper infestations, and CARMA fills that need. In the process CARMA provides advice regarding grasshopper population management options in an environmentally and economically sound fashion, and is the only pest management software that includes the more environmentally-friendly Reduced Agent-Area Treatments (RAATs) as a treatment option and an open-ended capacity for user-based treatment updates. This paper describes the most recent changes to CARMA with particular attention to the new architecture which demonstrates an approach to integrating an artificially intelligent LISP reasoner with a Java graphical user interface (GUI) in a way which combines the strengths of the two languages (i.e., LISP for artificial intelligence and Java for graphical user interfaces) in order to provide a strong reasoner while at the same time producing an appealing user interface which is platform independent and web capable.

Keywords: decision support, case-based reasoning, architecture, grasshopper pest management, LISP, Java

1 Introduction

CARMA is a decision-support system for grasshopper infestations that has been successfully used since 1996 (Hastings et al. 2002, Branting et al. 2001). CARMA demonstrates an approach to providing advice concerning the behavior of a complex biological system by exploiting multiple, individually incomplete, knowledge sources (Hastings et al. 1996) including utilization of a technique known as model-based adaptation which integrates case-based reasoning with model-based reasoning (Branting et al. 1997, Hastings & Branting 1995, Hastings et al. 1995, Branting & Hastings 1994).

Originally developed for rangeland grasshopper infestations in the state of Wyoming, in 2003 CARMA was extended with a cropland grasshopper advising module comprised mainly of a set of expert advising rules and a corresponding user interface (Hastings et al. 2003). Most recently, CARMA has undergone several changes including an extension of its advising abilities beyond Wyoming to include the states of Nebraska and South Dakota, a revamping of the graphical user interface, and a change to the underlying architecture.

This paper describes the most recent changes to CARMA with particular attention to the new architecture which demonstrates an approach to integrating an artificially intelligent LISP reasoner with a Java graphical user interface (GUI) in a way which combines the strengths of the two languages (i.e., LISP for artificial intelligence and Java for graphical user interfaces) in order to provide a strong reasoner while at the same time producing an appealing user interface which is platform independent and web capable.

Section 2 describes the need for a grasshopper infestation advisor. The environmental contributions of CARMA are set forth in Section 3. Section 4 provides a overview of CARMA's consultation process. Section 5 describes improvements to CARMA's interface along with the corresponding architectural challenges and modifications.

2 The Need for a Grasshopper Infestation Advisor

The need for accurate and reliable advice on the management of rangeland grasshoppers appears to be substantial and growing in the western states. Inflation and the restructuring of the USDA cost-share program have tripled the cost of rangeland grasshopper management for ranchers, compared with the control cost during the late-1980s. Most states no longer subsidize grasshopper programs; therefore, the cost of traditional tactics usually will exceed the benefits. Even though certain federal or state subsidies for grasshopper control are becoming available in some states, the expenses of a management program remain largely a producer's burden because of high insecticide and treatment costs. Rising treatment costs coupled with shrinking rangeland profit margins increasingly demand accurate selection of the most cost-effective responses to grasshopper infestations.

CARMA is a decision-support system which addresses the need for proper grasshopper infestation response by giving advice about the most economical and environmentally benign responses to grasshopper infestations based on 140 combined years of entomological expertise shared among experts in the 17 western states. As such, CARMA's grasshopper forage consumption predictions and treatment recommendations very closely approximate those of the experts. CARMA gives advice by comparing the current infestation to previous infestations (i.e., cases) and adapting recommendations accordingly. Infestation probabilities and treatment efficacies are used to predict re-infestations (Branting et al. 1997, Zimmerman et al. 2004); statistical methods are used to predict the economic benefits; and rules are used to select the treatments. Currently the geographic extent of CARMA covers grasshopper infestations in Wyoming, Nebraska, and South Dakota.

In addition to rangeland infestations, grasshoppers also routinely invade cropland, prompting a similar need for proper and accurate advice. In 2003, as a result of the collaboration between scientists from the University of Wyoming and the University of Nebraska-Kearney, a special module dealing with grasshopper infestations in cropland was developed for CARMA. The crop module handles several situations where grasshopper populations can develop at the rangeland-cropland interface including the frequently encountered situation where grasshopper infestations spread from rangeland into cropland, especially small grains. In addition, the module addresses grasshopper emergence within cropland, a scenario which has become more prevalent as of late with the adoption of "no-till" practices which create favorable conditions for grasshopper breeding within crops. Depending on the severity of an infestation and its potential to spread further into cropland, CARMA classifies cropland infestations according to economic damage potential and provides an explanation as to the source of the cropland grasshoppers (e.g., "Grasshoppers that prefer grasses or legumes are moving from the range through broadleaf weeds in the fence row area to get to vulnerable crops"). The addition of the crop module represents one more step in CARMA's continuing evolution to provide advice based on the most up-to-date pest management strategies. It is also an important asset for the potential expansion of the CARMA's geographic coverage beyond Wyoming, Nebraska and South Dakota.

During Grasshopper Train-the-Trainers workshops in 2003 through 2006 in Nebraska, Colorado North Dakota, South Dakota, New Mexico, Oregon and Montana, as well as during public meetings on grasshopper control in several Wyoming counties, land managers and pest control specialists expressed a great interest in CARMA. The attendees of these meetings were provided with free CD-ROMs containing updated versions of CARMA with the crop module. CARMA's demonstration by the University of Wyoming scientists was received with great attention during the Annual Meeting of the National Grasshopper Management Board in January 2004. Currently, CARMA's capabilities are being expanded to cover Montana and North Dakota, in addition to Wyoming, Nebraska and South Dakota which were covered before.

3 Environment Contributions of CARMA

CARMA provides the end-user with advice regarding grasshopper population management options in an environmentally and economically sound fashion. Historically, rangeland infestations were considered treatable when grasshoppers occurred at densities of eight or more grasshoppers per square yard. While this treatment threshold was thought to make sense from a protectionist point of view (i.e., protect the existing forage at all costs so as not to risk forage shortages), it did not always make economic sense (Lockwood & Schell 1995). CARMA conducts detailed analysis of infestations looking at a number of factors including grasshopper densities as well as range productivity in order to provide an economic analysis of an infestation. In cases where treatment costs will outweigh the estimated value of forage saved by treatment, CARMA advises a "no treatment" option, which provides the greatest environmental savings of all.

In addition to conventional, blanket applications of broad-spectrum insecticides like malathion and carbaryl, CARMA considers an option called Reduced Agent and Area Treatments (RAATs) (Lockwood & Schell 1997). RAATs is a method of integrated pest management (IPM) for rangeland grasshoppers in which the rate of insecticide is reduced from conventional levels as untreated swaths (refuges) are alternated with treated swaths. RAATs works both through chemical control, meaning grasshoppers are killed in treated swaths and as they move out of untreated swaths, and conservation biological control, which allows predators and parasites preserved in untreated swaths to suppress grasshoppers.

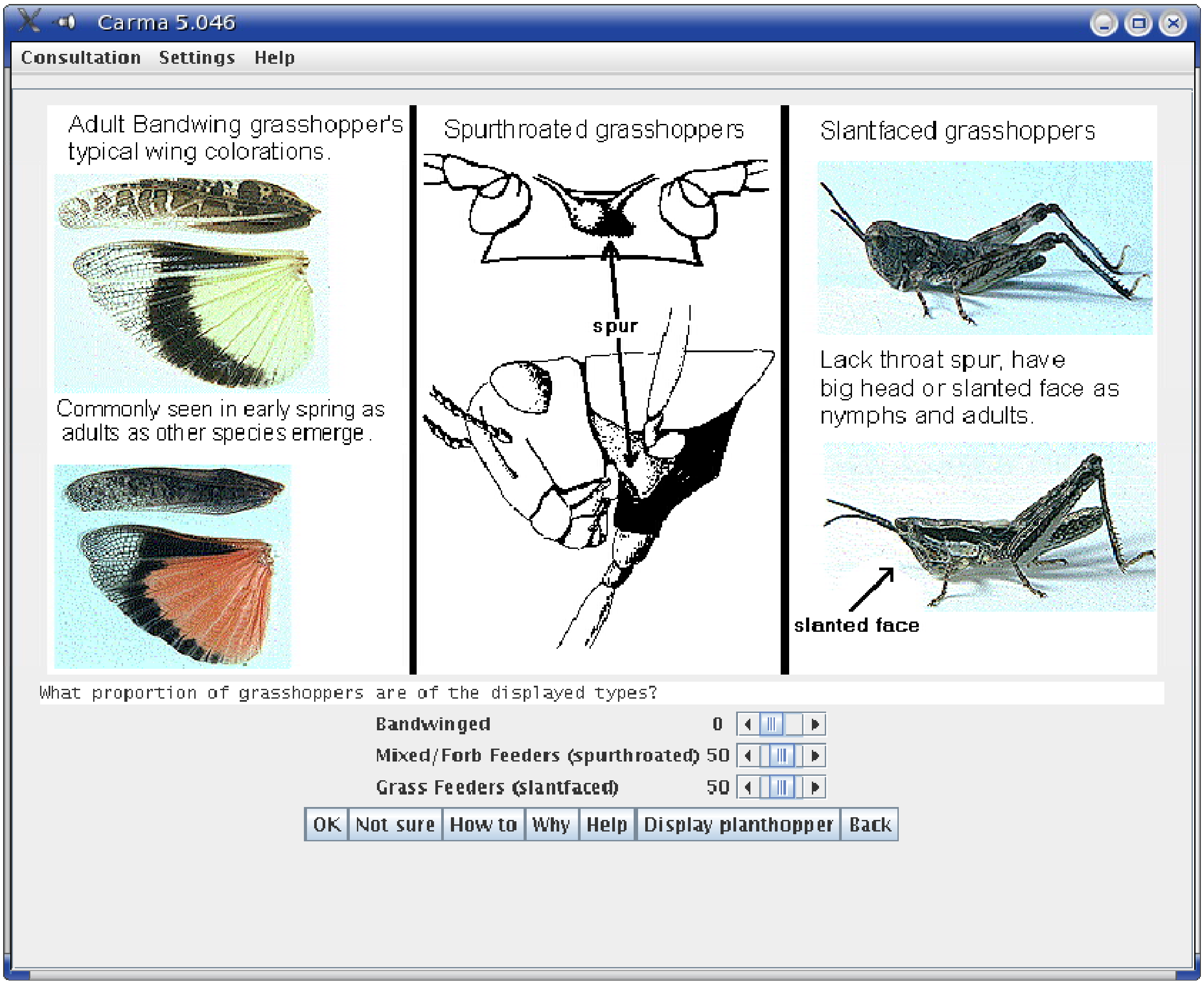

Figure 1: Elicitation of Grasshopper Type Information in CARMA.

Less insecticide in the environment lowers the risk to native species (including fish and wildlife), water quality, and humans. The untreated swaths provide a refuge for organisms with lower mobility than grasshoppers, and even those organisms that move into the treated swaths will be largely unaffected unless they feed on the foliage. The untreated swaths harbor species essential to rangeland ecosystems, including bio-control agents of grasshoppers and weeds. Low densities of surviving grasshoppers allow predators and parasites in the untreated refuges to re-colonize and thereby reestablish natural regulation of grasshopper populations. For these reasons, RAAT programs also may sustain higher densities of birds than blanket applications.

This IPM approach (RAATs) can reduce the cost of control and the amount of insecticide applied to our rangelands from 50 to 75% (Lockwood et al. 2002). RAATs is the preferred option in the USDA-APHIS EIS when grasshopper control is required (USDA 2002). CARMA is the only pest management software that includes RAATs as an option and an open-ended capacity for user-based treatment updates.

4 CARMA's Consultation Process

CARMA's primary task is to help end users (e.g., ranchers, land managers, pest control specialists) determine the most cost-effective responses to rangeland grasshopper infestations. Determining the most cost-effective responses requires, in part, estimating grasshopper forage consumption. Producing such an estimate requires predicting the behavior of a rangeland ecosystem. Because the complexity of the domain prevents precise modeling and simulation, CARMA's objective is to closely emulate grasshopper pest management experts who utilize a reasoning approach which is accurate, fast, explainable, and tolerant of inaccuracies in data. In predicting forage consumption, experts appear to use a form of case-based reasoning (CBR) by comparing new cases to prototypical cases, and adapting the consumption previously seen in the prototypical cases to account for any differences. For example, if the population density of emerging grasshoppers in a cool, wet spring is high (rather than moderate), an expert might predict moderately low (rather than low) forage consumption because a higher density generally means more consumption.

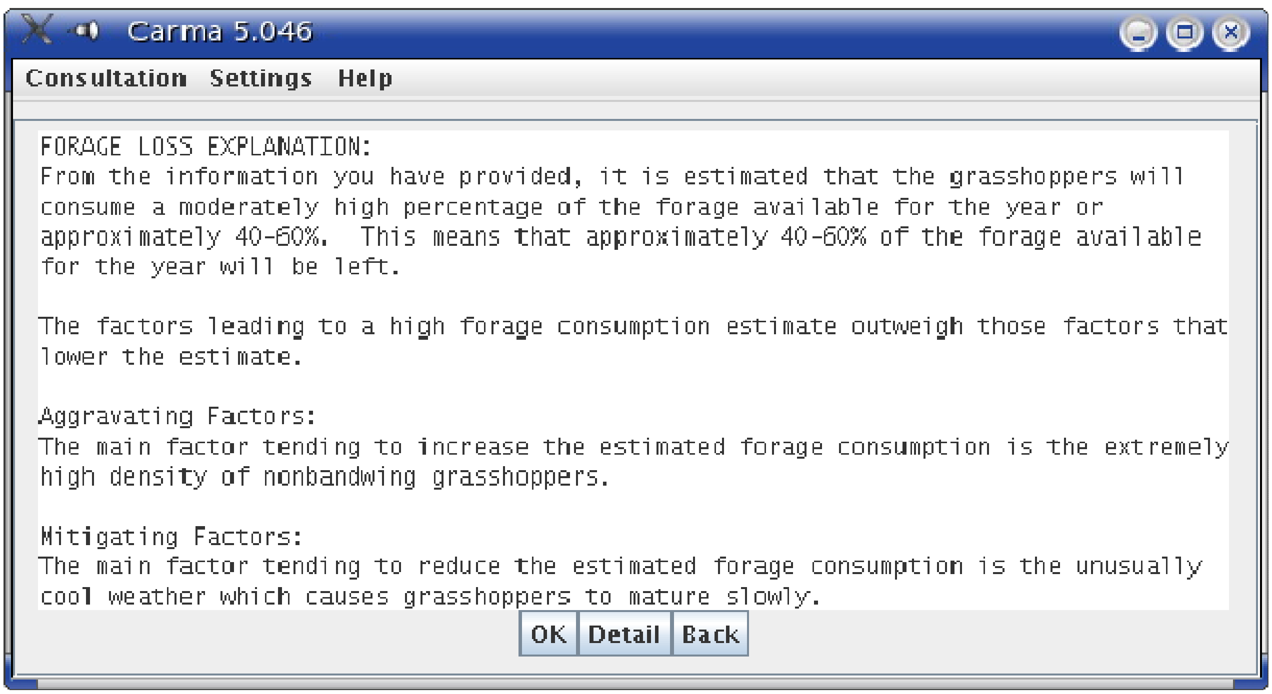

Figure 2: Forage Loss Estimation in CARMA.

CARMA's consultation process consists of the following steps:

1. Determine the relevant facts of the infestation case such as the species, population density, and developmental phases of the grasshoppers from information provided by the user by means of heuristic rules. CARMA uses multiple levels of rules for inferring each case feature, ordered by a qualitative estimate of each rule's accuracy or reliability. The rules are applied in succession until either the user can provide the necessary information or a default rule is reached. A typical interface window for determining the observed grasshopper type distribution appears in Figure 1. This interface window illustrates CARMA's revamped interface which will be described in Section 5.
2. Estimate the proportion of available forage that will be consumed by each distinct grasshopper population (i.e., subcase) by matching and adapting the prototypical infestation cases that best match the facts of the current case. The consumption predictions for each subcase are summed to produce an overall consumption estimate which is presented to the user along with an explanation of the estimated forage loss including the aggravating and mitigating factors (i.e., factors tending to increase or reduce the estimated forage loss, respectively). Figure 2 presents a typical window explaining estimated forage loss.

3. Compare total grasshopper consumption with the proportion of available forage needed by livestock. If the proportion of available forage that will be lost to grasshoppers and the proportion needed for livestock (and wildlife) exceeds 100% of the forage available, CARMA concludes that grasshoppers will cause economic losses.
4. If the predicted forage consumption will lead to economic loss, determine what possible treatment options are excluded by the case conditions (e.g., "Will it be hot at the time of treatment?" If so, exclude malathion).
5. Provide an economic analysis for each viable treatment option (including "no treatment") by estimating both the first-year and long-term savings. From the estimated savings, CARMA recommends the treatment or treatments that are most economical. CARMA also produces estimates of future reinfestation for each treatment option. A typical treatment recommendation window including estimates of future reinfestation and economic savings appears in Figure 3. CARMA lists both worst- and best-case scenarios for most calculations. Note that the analysis includes "no treatment" as an option and that negative savings indicate a loss.

Following a consultation, the user can then save and/or rerun the consultation with one or more case facts or treatment parameters altered. For example, the user can vary insecticide coverage and application rates to determine the best return on investment, and perhaps negotiate a price for a pesticide that is economical given the current infestation level and price of substitute forage. For a more detailed description of the rangeland grasshopper infestation advising task and the implementation of the consultation process within CARMA, the reader is referred to Hastings et al. (2002).

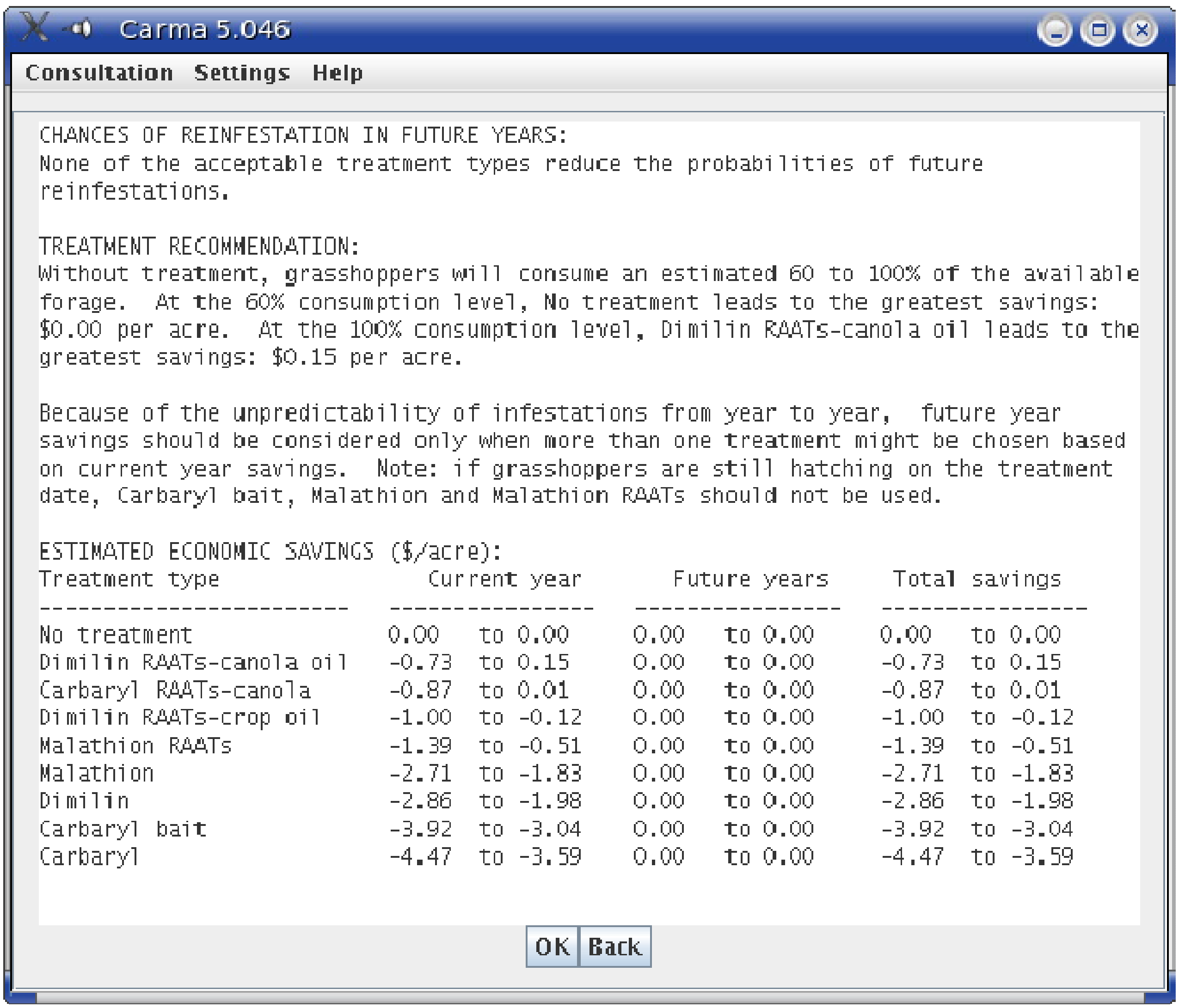

Figure 3: Treatment Recommendation and Economic Analysis in CARMA.

5 A Facelift for CARMA

LISP (short for List Processing) is a programming language which was invented in 1958 (Russell & Norvig 2003). LISP was the dominate artificial intelligence (AI) application language for approximately 25 years, and is still the favorite amongst AI enthusiasts due in part to its ability to support elegant solutions to problems which are too complex to solve with conventional programming languages. LISP programs generally run within a LISP interpreter rather than being compiled to executable code which runs native within a specific environment. A variety of different and not entirely compatible LISP flavors were combined in 1992 to produce ANSI Common LISP (Steele 1990). The core of CARMA was originally written in Common LISP and was capable of running on a wide variety of platforms (e.g, Linux, Macintosh, Windows) for which a LISP interpreter was available. A no cost, open-source LISP interpreter, GNU Common LISP (GNU 2007), was available at the time and many more have been developed since. Although the core of CARMA was capable of running on multiple platforms, the user interface was capable of running only on Windows through a proprietary and expensive Windows-specific LISP interpreter. While CARMA operated nicely for several years, problems began to pop up as the Windows operating system changed, in turn leading to changes in the proprietary LISP interpreter, and finally requiring changes to CARMA. Avoiding this maintenance expense was combined with our desire to field a product which was available to a wider user base than just Windows users. Common sense, then, dictated an approach not restricted to a single platform and not tied to an expensive solution (i.e., a Windows-specific LISP interpreter) for which CARMA would continue to face periodic, expensive interpreter upgrade fees and time consuming updates in order to continue operating. The challenge was thus to develop an approach which would make CARMA accessible on a variety of computing platforms while at the same time modernizing the interface and moving away from the expense of the proprietary LISP interpreter.

5.1 Architectural Challenges

Several approaches to revamping CARMA's architecture were considered. Each of the solutions include consideration of Java (Gosling et al. 2000), an interpreted programming language which has superb graphical capabilities and is quite platform independent due to the availability of Java interpreters (called Java Virtual Machines or JVM's by Sun Microsystems) for a wide variety of operating systems including PDA's.

1. **Complete rewrite in Java.** A complete conversion of the core reasoner and user interface to Java would provide platform independence of the resulting application, but given that the CARMA LISP core contained approximately 10,000 lines of code, and LISP and Java are entirely different languages, a complete rewrite would be a time consuming, if not almost overwhelming, task.
2. **Server-side LISP.** An Apache server module called mod_lisp (Battyani 2007) would support serving the CARMA core through Apache and require working out a corresponding web interface (e.g., using Java). An advantage would be that the core LISP reasoner would likely require minimal changes and additions in order to link with the user interface. Another advantage would be that CARMA could be updated as needed (e.g., to stay in line with current grasshopper management practices or to incorporate new data) and the user would simply access the CARMA web site in order to run CARMA rather than downloading CARMA to gain access to the most current version. The disadvantages include a dependence on the uptime of the server, and an inability to install and run CARMA locally (a big disadvantage for users who access CARMA in the field or in remote locations).
3. **A LISP interpreter written in Java.** A LISP interpreter written in Java would be capable of running the core of CARMA and could be written to so as to provide access from the LISP code to Java's graphical capabilities. A disadvantage of this approach is that the CARMA core would in effect be a program running inside of a LISP interpreter running inside a Java interpreter, and could suffer in terms of execution speed. Another disadvantage of this approach is that writing such a LISP interpreter (in Java) would be a daunting task. As with the previous approach, an advantage would be that the core LISP reasoner would require few changes. Additional advantages would include support for a modern user interface, and platform independence (i.e., CARMA could run on any platform for which a Java interpreter is available). In addition, by being wrapped inside a Java-coded interpreter and not tied to a particular platform, CARMA's code would be isolated from changes to operating systems (e.g., Microsoft) A further advantage of this approach is that CARMA would also be capable of running either as a user-installed application or through a web browser using Java Web Start, a technology which allows a user to easily access and run the most current version of a full-featured remote application.

Simple prototypes for the first two options were developed, but were deemed unacceptable based in large part on the disadvantages previously mentioned. The decision was made to proceed with the third option and in 2003, we began writing a LISP interpreter in Java. The effort was admittedly slow. Fortunately, someone beat us to the punch. Armed Bear Common LISP (ABCL) (Graves 2007) is a very capable Common LISP implementation that runs within a Java interpreter. Given the availability of ABCL, attention turned to developing the Java interface and linking it to the CARMA core through ABCL. The approach used for the new architecture of CARMA is described in the next section.

5.2 New CARMA Architecture

A general overview of CARMA's new architecture appears in Figure 4. CARMA is comprised of two distinct parts:

1. A Java graphical user interface which runs as a Java application inside the interpreter (JVM), and

2. The LISP core reasoner which runs as a LISP application inside the ABCL interpreter (which in turn runs inside the JVM).

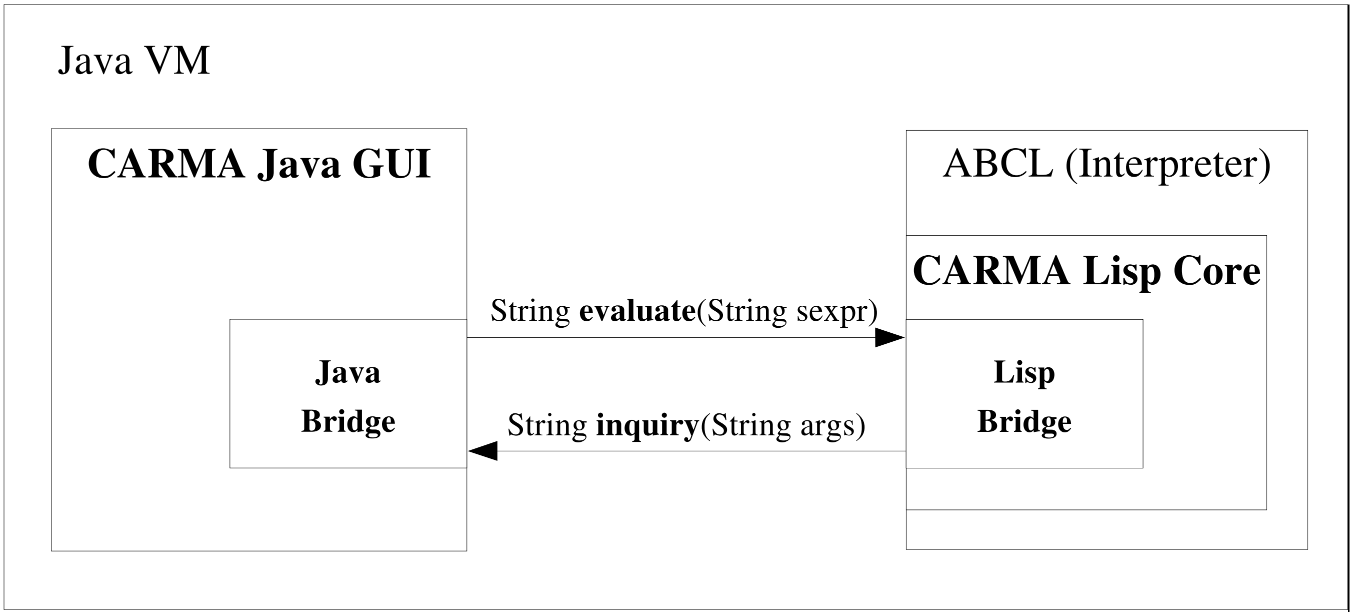

Figure 4: A general overview of the architecture of CARMA.

Two way communications between the two components is accomplished by bridge modules. In general, the Java bridge is responsible for initializing the ABCL interpreter and asking the ABCL interpreter to evaluate LISP s-expressions (i.e., calling LISP functions and receiving back results). The LISP bridge directs questions and messages to the Java bridge whenever such information should be presented to the user through the GUI. A brief overview of this communication process within CARMA is as follows:

1. During initialization of the CARMA Java GUI, the Java bridge forces loading of the ABCL interpreter by evaluating a dummy expression through the statement

   ```
   Interpreter.evaluate("(+ 3 5 )" );
   ```

   and follows by loading each of the CARMA LISP files into the LISP interpreter.

2. Once the Java GUI has finished initialization and is available, the user can select from a number of different menu options. The main (but not only) function of the GUI is to allow a user to run an advising consultation. When the user instructs the GUI to run a consultation, the Java bridge tells the LISP interpreter to process a new consultation by calling

   ```
   Interpreter.evaluate("(consultation nil)" );
   ```

3. When the LISP bridge is instructed to run a consultation, it initializes the LISP to Java communication path as follows:

   (a) If a Java bridge object has not yet been created [1], it is created using the ABCL methods `jnew` and `jconstructor`:

   ```
   (setq *java-bridge-object* (jnew (jconstructor "Bridge")))
   ```

   (b) Links to Java bridge methods are made using the ABCL method `jmethod`, *e.g.*,

   ```
   (setq *java-bridge-inquiry-method*
       (jmethod "Bridge" "inquiry" "java.lang.String"))
   ```

[1] The LISP bridge is responsible for creating the Java bridge object to which it will direct its communications. Prior to this point, the Java bridge code is executed at the static class level.

4. Once the LISP to Java bridge is initialized, the consultation begins. During its reasoning process, the LISP core will direct questions or messages to the Java bridge:
   (a) When the LISP core has a question for the user, it passes the question along with any additional arguments as a string (e.g., "location-state" for determining the state of the infestation) to the Java bridge through the LISP bridge which prompts the appearance of an input window. A typical CARMA Java GUI input window appears in Figure 1. The user's response is passed back to the LISP bridge in the form of a string:

```
(setq return-string (jcall *java-bridge-inquiry-method*
    *java-bridge-object* question))
```

   (b) On the Java bridge side, the method signature looks like

```
public String inquiry (String args)
```

   (c) The LISP core parses the returned string into a list and uses the result in the analysis of the consultation.

5.3 Minor Development Issues

For the most part, the challenges encountered during the development of the new architecture mostly related to determining how to make the bridge modules communicate properly. In general, very few lines of code within the CARMA LISP core were touched other than rewriting the code which previously called the Windows-specific LISP GUI code. Once the new architecture was complete, CARMA ran into some problems with Java stack overflow errors. Java is not as adept at handling recursion. However, CARMA, like many LISP programs, makes extensive use of recursion as a looping mechanism. CARMA's recursion pointed out a slight weakness in the Java (or ABCL) handling of recursion. Rather than trying to potentially eliminate the issue by increasing the Java stack size or further looking into the ABCL code, just a few recursive LISP functions were rewritten until a CARMA advising consultation could run to completion without errors on a personal computer with limited memory. It is possible that this issue could reappear on less powerful computing devices such as PDA's.

Another slight issue was the embedding of ABCL within CARMA. The most recent version of ABCL, 0.0.9, contains compiled LISP modules which create a larger distribution size and longer download times. For this reason, CARMA uses ABCL 0.0.8 which has LISP modules in plain text LISP source code format. Although this choice might require a slightly longer time for loading the LISP interpreter (if LISP source modules do indeed take longer to load than compiled LISP modules), the time needed for downloading CARMA is reduced.

5.4 Availability

The most recent version of CARMA, 5.046, with grasshopper advising capabilities for Nebraska, South Dakota and Wyoming is available free of charge for noncommercial purposes and can be downloaded and installed from http://carma.unk.edu or run as a Java Webstart application.

6 Summary

CARMA is a decision-support system for grasshopper infestations that has been successfully used since 1996. Rising treatment costs coupled with shrinking rangeland profit margins increasingly demand accurate selection of the most cost-effective responses to grasshopper infestations, and CARMA fills that need. In the process CARMA provides advice regarding grasshopper population management options in an environmentally and economically sound fashion, and is the only pest management software that includes the more environmentally-friendly Reduced Agent-Area Treatments (RAATs) as a treatment option and an open-ended capacity for user-based treatment updates. The most recent changes to CARMA's architecture demonstrate an approach to integrating an artificially intelligent LISP reasoner with a Java graphical user interface (GUI) in a way which combines the strengths of the two languages (i.e., LISP for artificial intelligence and Java for graphical user interfaces) in order to provide a strong reasoner while at the same time producing an appealing user interface which is platform independent and web capable.

7 Acknowledgments

CARMA's development since 2003 was supported through funds from Cooperative Agreements between USDA-APHIS-PPQ (Western Region) and the University of Wyoming (grants USDAAPHIS5112 and USDAAPH44906).

References

Battyani, M. 2007. Fractal Concept: mod_lisp home page (web page). http://www.fractal concept.com/asp/mod_lisp (Accessed 1 March 2007).

Branting, L. K., and Hastings, J. D. 1994. An empirical evaluation of model-based case matching and adaptation. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94) Workshop on Case-Based Reasoning (WS–94–01)*, 72–78. Seattle, Washington: AAAI Press.

Branting, L. K.; Hastings, J. D.; and Lockwood, J. A. 1997. Integrating cases and models for prediction in biological systems. *AI Applications* 11(1):29–48.

Branting, K.; Hastings, J.; and Lockwood, J. 2001. CARMA: A case-based range management advisor. In *Proceedings of The Thirteenth Innovative Applications of Artificial Intelligence Conference (IAAI-01)*, 3–10. Seattle, Washington: AAAI Press.

GNU. 2007. GCL - GNU Common Lisp (web page). http://www.gnu.org/software/gcl (Accessed 1 March 2007).

Gosling, J.; Joy, B.; Steele, G.; and Bracha, G. 2000. *The Java Language Specification*. Boston, MA: Addison-Wesley, 2nd edition.

Graves, P. 2007. Armed Bear Common Lisp (web page). http://armedbear.org/abcl.html (Accessed 1 March 2007).

Hastings, J. D., and Branting, L. K. 1995. Global and case-specific model-based adaptation. In *Proceedings of the AAAI 1995 Fall Symposium on Adaptation of Knowledge for Reuse (FS–95–04)*, 47–53. Cambridge, Massachusetts: AAAI Press.

Hastings, J.; Branting, K.; Lockwood, J.; and Schell, S. 2003. CARMA+: A general architecture for pest management. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03) Workshop on Environmental Decision Support Systems (EDSS-03)*, 18–21.

Hastings, J. D.; Branting, L. K.; and Lockwood, J. A. 1995. Global and case-specific model-based adaptation. In *Proceedings of the First International Conference on Case-Based Reasoning (ICCBR-95) Lecture Notes in Artificial Intelligence 1010*, 181–192. Sesimbra, Portugal: Springer.

Hastings, J.; Branting, K.; and Lockwood, J. 1996. A multi-paradigm reasoning system for rangeland management. *Computers and Electronics in Agriculture* 16(1):47–67.

Hastings, J.; Branting, K.; and Lockwood, J. 2002. CARMA: A case-based rangeland management adviser. *AI Magazine* 23(2):49–62.

Lockwood, J. A., and Schell, S. P. 1995. Rangeland grasshopper outbreak dynamics: gradient, eruptive, both, or neither? *Journal of Orthoptera Research* 4:35–48.

Lockwood, J. A., and Schell, S. P. 1997. Decreasing economic and environmental costs through reduced area and agent insecticide treatments (RAATs) for the control of rangeland grasshoppers: Empirical results and their implications for pest management. *Journal of Orthoptera Research* 6:19–32.

Lockwood, J. A.; Anderson-Sprecher, R.; and Schell, S. P. 2002. When less is more: optimization of reduced agent-area treatments (RAATs) for management of rangeland grasshoppers. *Crop Protection* 21:551–562.

Russell, S., and Norvig, P. 2003. *Artificial Intelligence: A Modern Approach*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2nd edition.

Steele, G. L. 1990. *Common Lisp: The Language*. Newton, MA, USA: Digital Press, 2nd edition.

USDA. 2002. Rangeland grasshopper and mormon cricket suppression program, Final environmental impact statement–2002. Technical report, U.S. Department of Agriculture, Animal and Plant Health Inspection Service, Riverdale, MD, USA.

Zimmerman, K. M.; Lockwood, J. A.; and Latchininsky, A. V. 2004. A spatial, markovian model of rangeland grasshopper (Orthoptera: Acrididae) population dynamics: Do long-term benefits justify suppression of infestations? *Environmental Entomology* 33(2):257–266.