Reprinted from

Computers and electronics in agriculture

Computers and Electronics in Agriculture 14 (1996) 291–300

Developing transportable agricultural decision support systems: Part 1. A conceptual framework

Gianni Jacucci [a], Mark Foy [a,*], Carl Uhrik [b]

[a] *Laboratorio di Ingegneria Informatica (LII), Dipartimento di Informatica e Studi Aziendali (DISA), Università degli Studi di Trento, Via Fortunato Zeni 8, 38068 Rovereto (TN), Italy*
[b] *University of Phoenix, DTP Faculty, 7800 E. Dorado Pl., Englewood, CO 80111, USA*

ELSEVIER

Computers and electronics in agriculture

GENERAL INFORMATION

Audience: All scientists, engineers, environmentalists and economists concerned with applications of computers and electronics to agriculture and related industries.

Aims and Scope: *Computers and Electronics in Agriculture* provides international coverage of advances in the application of computer hardware, software and electronic instrumentation and control systems to agriculture, forestry and related industries. The latter include horticulture (in both its food and amenity aspects), forest products, aquaculture, animal/livestock science, veterinary medicine and food processing.

The journal publishes original papers, reviews, applications notes and book reviews on topics including computerized decision-support aids (e.g., expert systems and simulation models) pertaining to any aspect of the aforementioned industries; electronic monitoring or control of any aspect of livestock/crop production (e.g. soil and water, environment, growth, health, waste products) and post-harvest operations (such as drying, storage, production assessment, trimming and dissection of plant and animal material). Relevant areas of technology include artificial intelligence, sensors, machine vision, robotics and simulation modelling.

Publication information: *Computers and Electronics in Agriculture* (ISSN 0168-1699). For 1996 volumes 14 and 15 are scheduled for publication. Subscription prices are available upon request from the Publisher. Subscriptions are accepted on a prepaid basis only and are entered on a calendar year basis. Issues are sent by surface mail except to the following countries where Air delivery via SAL mail is ensured: Argentina, Australia, Brazil, Canada, Hong Kong, India, Israel, Japan, Malaysia, Mexico, New Zealand, Pakistan, PR China, Singapore, South Africa, South Korea, Taiwan, Thailand, USA. For all other countries airmail rates are available upon request. Claims for missing issues should be made within six months of our publication (mailing) date. Please address all your requests regarding orders and subscription queries to: Elsevier Science B.V., Journal Department, P.O. Box 211, 1000 AE Amsterdam, The Netherlands, tel. (+31-20)4853642, fax (+31-20)4853598.

In the USA and Canada: For further information on this and other Elsevier journals please contact: Elsevier Science Inc., Journal Information Center, 655 Avenue of the Americas, New York, NY 10010, USA. Tel. (212) 6333750; fax (212) 633 3764; telex 420-643 AEP UI.

US mailing notice: *Computers and Electronics in Agriculture* (ISSN 0168-1699) is published monthly except in February, May, July and October (total 8 issues) by Elsevier Science (Molenwerf 1, P.O. Box 211, 1000 AE Amsterdam). Annual subscription price in the USA is US$500.00 (valid in North, Central and South America), including air speed delivery. Second class postage paid at Jamaica NY 11431.
USA POSTMASTER: Send address changes to Computers and Electronics in Agriculture (ISSN 0168-1699), Publications Expediting Inc., 200 Meacham Avenue, Elmont, NY 11003.
AIRFREIGHT AND MAILING in the USA by Publications Expediting Inc, 200 Meacham Avenue, Elmont, NY 11003.

Back volumes: Please contact the Publisher

ELSEVIER Computers and Electronics in Agriculture 14 (1996) 291–300

Computers and electronics in agriculture

Developing transportable agricultural decision support systems: Part 1. A conceptual framework

Gianni Jacucci [a], Mark Foy [a,*], Carl Uhrik [b]

[a] Laboratorio di Ingegneria Informatica (LII), Dipartimento di Informatica e Studi Aziendali (DISA), Università degli Studi di Trento, Via Fortunato Zeni 8, 38068 Rovereto (TN), Italy
[b] University of Phoenix, DTP Faculty, 7800 E. Dorado Pl., Englewood, CO 80111, USA

Accepted 7 September 1995

Abstract

Developers of agricultural decision support systems (DSSs) have had little guidance in constructing DSSs which could be widely used. To assist, a conceptual framework is introduced which proposes methods for making DSSs transportable so that their utility is high. This framework serves as a checklist and includes recommendations about general implementation, user interaction, data management, and models. General implementation aspects emphasize portable and public domain tools, and user interaction aspects the use of a graphical user interface (GUI); flexible data management systems play a large role in satisfying the suggested data management aspects; and an intelligent model adaptation strategy is one of the main model aspect recommendations.

Keywords: Decision support system; Modeling; Portability; Data management; User interface

1. Introduction

In agriculture, the utilization of decision support systems (DSSs) throughout a broad geographical area (i.e., inter-regional transportability of DSSs) is often hindered by one or more problems including:

(a) *hardware platform* (i.e., running the DSS on the computer in the new location);

(b) *data compatibility* (i.e., using the data as stored in the new location within the DSS and its included models);

(c) *accuracy* (i.e., obtaining accurate output from the models included in the DSS when it is run in the new location).

These are some of the major difficulties of *DSS/model technology transfer*. One

* Corresponding author. Present address: 1818 19th Street No. 309, Boulder, CO 80302-5563, USA.

SSDI 0168-1699(95)00039-9

of the solutions to overcoming these difficulties is to build transportable DSSs which can be moved between regions, thereby allowing wide DSS dissemination and utilization. To give guidance to DSS developers, a conceptual framework for assisting these developers in creating transportable DSSs is proposed in this paper.

The framework is broken into four aspects: (1) general implementation; (2) user interaction; (3) data management; and (4) model aspects, and each has several prescriptions. The general implementation aspects emphasize low-cost portability. The user interaction aspects focus on ease-of-use and flexibility. The data management aspects highlight the importance for the DSS to handle many types of data. The model aspects stress that the models should be adaptable, and that mechanisms should be in place to help the user perform this adaptation.

First, in Section 2, the four main aspects of this framework (general implementation; user interaction; data management; and model aspects) will be discussed. Following from this, in Section 3, some advantages and disadvantages of this framework will be outlined. Lastly, conclusions are given.

2. Description of the framework

The framework presented here is designed to guide a DSS developer in the creation of transportable DSSs. This development framework emphasizes creating DSSs which are easily disseminated and transported in both the development stage and in the user distribution stage. All of the aspects listed below relate in some way to transportability. Fig. 1 illustrates the four main aspects (and sub-aspects) of the transportable DSS framework proposed here.

The following discussion will be divided into four sub-sections: (1) general implementation aspects; (2) user interaction aspects; (3) data management aspects; and (4) model aspects. The general implementation aspects will include basic DSS development recommendations regarding programming tools (public domain and

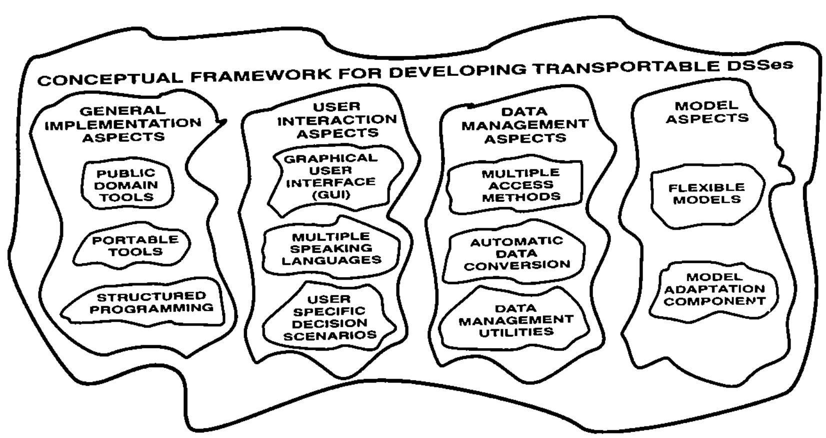

Fig. 1. Aspects of the conceptual framework for developing transportable DSSs.

portable tools), programming structure, and user interface [graphical user interface (GUI) and speaking languages]. The data management aspects will emphasize the need for robust data management, and last, the model aspects will focus on the importance of adaptability of models to different user needs and different locations.

2.1. General implementation aspects

2.1.1. Public domain tools

The use of public domain tools to design and build DSSs have some distinct advantages and disadvantages over commercial tools. The primary advantages of public domain tools stem from cost. By choosing tools which co-developers and eventual users will not have to pay for (or will not have to pay much for) is very appealing because it makes it easier to achieve wider distribution of the DSS. Another advantage is that with public domain tools, source code is often available, so it is possible for the developer to customize and change the package to fit her or his needs. Additionally, the final DSS may be modified by recipient users if they have the knowledge and desire to make changes. The primary disadvantages have to do with the usability and functionality of public domain software. That is, sometimes public domain software is "buggier" and less sophisticated than commercial software, so it can limit a project more than commercial software might. Overall, this framework advocates public domain software because of the great dissemination advantage it provides.

2.1.2. Portable tools

Portable computer tools (i.e., tools which can be used on multiple computer platforms) can be used in the development of DSSs and models to insure that the results are transportable between various common computer platforms. This is recommended because different groups have different preferences in computing environments, and it is not desirable to limit the potential users of the system just because some do not have the "right" type of hardware. This is important both for the development stage and for the dissemination stage.

2.1.3. Structured programming

To make any system extendible and easily modifiable, the code should be modular and consistently commented, indented, and structured. Hence, program guidelines are particularly essential for the development stage if the DSS and/or models within the DSS are to be moved between programmers during development. As well, modularity permits developers to "turn-off" different parts of the system as needed (during development or dissemination). A commonly understood structure and design will allow programmers to work separately and bring the code together when they want to create a combined system (particularly in the end to create a final product). Therefore, a common program structure and design should be agreed upon relatively early, especially before existing models are modified or new models are added to the DSS.

2.2. User interaction aspects

2.2.1. GUI

A key aspect for an agriculturalist's acceptance (or rejection) of a DSS is how easy the system is to use. By presenting the DSS (and the models inside) through a GUI at the top level, the likelihood of successful system transportation/dissemination increases. Overall, by presenting the user with information in a friendly, graphical, non-redundant, consistent manner, the system is likely to be used more often because it is not threatening, and it is easy and pleasant to use.

2.2.2. Multiple speaking languages

The last "general" aspect for the development of transportable DSSs prescribes the inclusion of multiple spoken languages so that text presented by the system to the user (outputs of models, etc.) can be presented in the language most preferred by the user. This is important if the system is to be developed and/or used in multiple countries with multiple spoken languages. With this functionality, the agriculturalist using the system will be able to better understand the information presented when it is given in her/his native language.

To make this goal easier to achieve and maintain, it is important to structure the program code such that these different spoken languages can be included in a modular way.

2.2.3. User specific decision scenarios

A DSS user's view of a problem is usually radically different from a model developer's view (Noell, 1992). The modeler is concerned with capturing a sufficient number of relationships in a model so as to permit comparisons between behaviors of the model and behaviors observable in the real world. However, DSSs are above all dedicated to solving users' problems, not making scientific studies. Thus, a model contained within a DSS is only a part of the problem-solving picture. The model must be effectively driven to provide a user with information that will assist in making decisions.

The DSS acts as a buffer between the user and the model, moderating the solicitation of relevant user inputs, executing the model, and subsequently interpreting and displaying the results that are relevant to the user problem-solving needs. Note that both inputs and outputs of the DSS must be relevant to the user's problem. Model inputs or outputs thus often require some sort of translation from or to *user specific data*. Consequently, DSSs typically contain one or more *user specific "scenarios"* that are problem specific in which one can identify the following distinct features:

(a) an expected set of inputs — some of which are "decision variables" directly modifiable by the user and some of which are relatively static (i.e., pre-assigned default values set by experts or values coming from databases/files outside the DSS);

(b) some strategy for driving the model, possibly through a search or evaluation of multiple alternatives; and

(c) an expected set of results in a format that is meaningful and preferably visually appealing.

Thus a DSS must be respectful of the particular user's needs, but usually this is a compromise: scenarios should be as generic as possible but allow for some level of customization. Incorporating multiple scenarios for the same model into a DSS provides some degree of flexibility. Additional flexibility is gained in allowing different options for inputs and for outputs.

Two simple ways of implementing this flexibility are applicable in almost every case. The first way is to use optional preference menus and tables or files to specify user specific configurations which describe what defaults to use, what sort of units to assume on input/output data, and what display formats the user prefers. A second (nearly universal) way to achieve flexibility is through utility programs for making conversions between data quantities and units (refer to the aspects of data management above). The reason that such utilities might be effectively "hidden" within the model is that sometimes it is undesirable or dangerous to make such conversions too "open" to direct influence by the end-user. In many cases, the model quantities will be completely irrelevant to the user and should thus remain hidden from the user's view (Fedra and Winklebauer, 1991).

Occasionally, more advanced approaches involving expert systems or rule bases may be useful so that further user specific custom tailoring of a scenario can be achieved. For inputs, converting circumstantially available user inputs (inputs that the user happens to have conveniently available) into data that will be used by the models can spare the user the effort and cost of collecting the actual data needed by the models which might not be so readily available. Analogously, for the outputs, rule-based systems can assist in interpreting model outputs on behalf of the user in order to address the user specific problems. Finally, in recognizing that the point of most DSS scenarios is to compare alternative situations, the basic problem-solving scenarios can effectively be extended by rules providing a systematic way of indicating how outputs from one situation should be manipulated to create an input for a new situation to be evaluated — for example, when a particular button is pushed from the interface, the output quantity X could be equally distributed to each of three inputs:

IF (button1) and (scenario1) and ($X > 0$) THEN

$$I1 = I1 + X/3,\ I2 = I2 + X/3,\ I3 = I3 + X/3.$$

The goal is to create the DSS with generic scenarios which are applicable to most users in a particular region, along with a generally applicable set of default preferences installed that reflect the desire of the majority of users in that region, but leave the individual users the option to customize at will. Moreover, although the agenda for the problem-solving scenario is usually preset by a user requirements analysis, there should remain room for multiple approaches to solving the problem on the part of the many diverse end-users. Alternatives should be compared and ranked, but a so-called "optimal" or "desirable" choice should be left to the user. The scenarios should be user relevant, but because "the user" is a multitude, there must remain generality within the scenarios.

2.3. Data management aspects

The availability and accessibility of data is usually the key to the functioning of a DSS because all DSSs need data. Therefore, the way in which models are provided with data (i.e., how models can request data and how models can elicit data from users) has a large effect on how well a DSS functions. This particularly applies to meteorological data in an agricultural scenario because most agricultural DSSs contain models which rely upon access to temporal meteorological data.

2.3.1. Multiple access methods

With the goal to make the final DSS transportable, a flexible and easily extensible data management system should be employed. This system should allow many different types of data to be used by the models (i.e., it should have multiple access methods). With a robustly designed system, models should not have to be modified or re-written to use different types of data, rather "generic" calls for data within the model should be used, and these calls should be compatible with data from many different sources.

The most straightforward way of implementing such functionality (for example, for handling meteorological data) is to create a set of procedures (e.g., a library) which handle the management of the data from a high level point of view without reference to low level details. That is, the low level access methods for storing and retrieving data (e.g., files, databases, network services, etc.) should be hidden within this library, and "generic" calls to the library should not make reference to the underlying scale of units, granularity or format of the data. Moreover, from the user's point of view, restrictions should be placed such that the user can only access the data through the "generic" functions. Thereby, the specification of the data source can be accomplished with one call to this library, and thereafter, when data is requested by the model (using the "generic" calls), data will be retrieved from whichever source of data is currently *open*. Then, if a different source of data is required, all that has to be done is to *close* access to the current data source, and *open* access to the new data source.

2.3.2. Automatic data conversion

Automatic conversion between data of varying granularity and units should be relatively invisible to the user. For example, it should make no difference whether data is recorded hourly or minutely, no more than whether precipitation is recorded in millimeters or centimeters. Since sometimes wholly new quantities could be required by a model (i.e., quantities different from those recorded by the user), more than just a linear scaling, temporal compression, or filling of data will be required to effect the necessary conversions. Complex non-linear formulas may be involved to convert between data that, though equivalent, are essentially different quantities (e.g., hours of daylight versus Joule/m^2 of radiation). Special procedures are nonetheless available for many such conversions. For examples the reader is referred to the FAO literature on irrigation, and in particular to Doorenbos and Pruitt (1977) and Verhoef and Feddes (1991).

2.3.3. Data management utilities

Another feature to make the DSS more attractive and useful is the addition of utilities for inputting, viewing and modifying data. Database managers and editors for this purpose become even more useful when integrated with the above mentioned procedures for invocation of conversions and cross-translating between data access methods, especially in order to create new data sets for purposes not imagined in the scope of the original DSS.

2.4. Model aspects

2.4.1. Flexible models

Flexible models allow the user, if desired, to change the way in which a model runs (i.e., change the modeling options). The inclusion of model options will especially benefit users who:

(1) do not have particular data;

(2) have preferences about which of several alternative data should be used; or

(3) have preferences about which optional steps to include in a model calculation.

Model options should make the model more transportable and robust, thereby allowing the model to be employed across a wider range of circumstances.

Overall, by changing a model to allow users modeling options, the model becomes usable by more people, each having their problem specific needs served through the selection of those options.

2.4.2. Model adaptation component

The construction of transportable DSSs/models is served by the inclusion of a model adaptation component which can adapt a model to a particular locality so that it will run well in the new location. Model adaptation may be necessary because often when models are transported from one region where they run accurately to a different region, they do not give accurate output (such as recommendations, results, and/or indicators) in their new environment (i.e., when they are run with the data of a new region).

Because it is best not to require significant effort and knowledge on the part of an agriculturalist, it is most desirable to have this adaptation done in an intelligent way which minimizes how much the user will have to intervene (and how much the user will have to know) to achieve this adaptation. Overall, it is best to minimize the number of items which a user must specify.

The general principle of allowing the user to adapt a model should be as follows:

(a) the user should enter some basic factual information (probably observed historical data such as crop yield or fungus infection dates in past years);

(b) the adaptation component should be activated with this information.

The user should not be required to enter additional interpolated information or "thought-out"/"processed" information.

An intelligent, automated component would be ideal in this case; thus, an artificial intelligence (AI) technique could be considered to fulfill this need. One particular methodology which utilizes an AI search technique called genetic algorithms (Goldberg, 1989), and fulfills the adaptation component criteria, is proposed in part two of this two-part paper series.

3. Advantages and disadvantages of this framework

There are some evident disadvantages of this framework which should be considered and weighed against the benefits before implementing a DSS under the framework described.

3.1. Advantages

The main potential benefits and advantages of using the above described framework which emphasizes the utilization of techniques to make DSSs transportable versus more platform or location specific techniques include:

(1) increased availability, usability, accessibility and distribution (i.e., by making a DSS transportable, the DSS will become more available, usable, and accessible in the future by a larger set of agriculturalists due to the fact that once the initial work has been done to make a DSS transportable, the work to transport the DSS to new regions should be reduced. Additionally, regions that do not have resources or expertise to create DSSs will have access to methods that would never have been available to them without transferable DSSs);

(2) increased ease in obtaining assistance in developing a DSS (i.e., more people can more readily contribute to the building of the DSS since it can be moved easily);

(3) increased DSS robustness (i.e., by applying effort to make a DSS transportable, the DSS is developed in a way that makes it more robust because there will be a conscious effort to eliminate all particularities which make a DSS location specific);

(4) increased DSS modularity, openness, and independence (i.e., by making a DSS transportable, the DSS becomes more modular, open, and independent because in building transportable DSSs, parameter settings must be easy to adjust; this allows expert agriculturalists to more easily explore, understand and modify the DSS, even allowing the expert to quickly make changes by hand for testing purposes, often without recompiling the DSS code);

(5) increased testability and reliability (i.e., by making the DSS usable with a wider set of data sources, it implicitly becomes more testable and can subsequently become more reliable).

3.2. Disadvantages

Potential problems and disadvantages of following the prescription of this framework include:

(1) avoidance of potentially high-utility environment specific tools (i.e., this framework does not recommend the utilization of environment specific tools and techniques which could in some situations significantly increase the overall worth of the system; e.g., if a computer platform offers a very nice, easy to implement user

interface design tool with many functions such as automated graph generation, but this tool cannot be used on other platforms, the framework described here would prescribe the avoidance of this tool due to its inability to be transported);

(2) risk of having "brittle" models which are not transportable (i.e., this framework assumes the models in the DSS are transportable, but they may not be, so the effort to make a transportable DSS may be wasted if an essential model is very difficult to make transportable);

(3) social rejection (i.e., it is possible that agriculturalists outside the region for which a DSS was developed will reject the DSS because they feel it is too general and they are not comfortable using such a general DSS which is not custom tailored to their needs, even if it has many useful functionalities);

(4) increased costs and efforts (i.e., generally, more resources are required to build a transportable DSS than a location specific DSS).

Because there are disadvantages to always utilizing *ALL* parts of this framework, the best advice to a DSS/model developer is to strive to work within the above framework as much as possible, but realize it could be greatly beneficial to "go against" the framework when significant increases in the worth of the system are achieved. For example, a developer may choose to use a *commercial* (rather than public domain) GUI builder tool because it has a number of advantages (e.g., portability, easy development, increased DSS attractiveness) if this developer thinks that users in the target market can afford to purchase any needed runtime licenses for the commercial tool when putting the developed system into use.

Additionally, it should be recognized that some parts of this framework will not be applicable to certain DSS development efforts. For example, the public domain, portability, and multiple speaking language recommendations may not be applicable to groups who are custom designing a DSS for a particular entity which will use and support the DSS solely as is after its implementation (i.e., the entity will not want to disseminate it for wider use and will never adapt it for use outside the narrow context for which it was originally intended). This does not mean that other parts of the framework (such as the model adaptation component) will not be useful to this entity.

Overall, because of the advantages, it is believed that most users will want to follow most of the guidelines prescribed in this framework so as to achieve the most transportable DSS/models.

4. Conclusion

The conceptual framework presented here addresses *DSS/model technology transfer* (i.e., the moving of functional and useful agricultural DSSs/models that are developed in one location to a new location so they can be used in this new location), focusing on four main recommendation aspects: (1) general implementation; (2) user interaction; (3) data management; and (4) model aspects. These aspects cover recommendations from the types of tools that should be utilized to create a transportable DSS, to data management techniques, to model adaptation requirements.

Advantages and disadvantages of utilizing this framework to create transportable DSSs have been outlined. It has been observed that applying *ALL* of the presented framework criteria is not always a good idea, and this should be evaluated by the developer in context; but the general conclusion is that transportability (whether it is done following this framework or some other) usually should be an important part of building DSSs so that this technology can be distributed to benefit larger groups of people.

The applicability of the framework may extend also into the initial development of models as well. For instance, it allows a model developer/programmer to concentrate on developing their model without having to consider the origin of the meteorological data. At least, in any case, an efficient field testing and dissemination of agricultural models would be promoted by the adaptation of pieces of the framework by more model developers.

In part two of this two-part paper series, a DSS which has followed the above framework guidelines will be presented. In creating this DSS, many design decisions have been made in the context of the framework's recommendations, including the design and implementation of two major components to satisfy part of the framework's data management and model aspects.

Acknowledgments

This work has been performed under a project called SYBIL. Funding has been provided for the SYBIL project (project number PL 900615) by the EC CAMAR Program (Competitiveness in Agriculture). The authors wish to thank Dr. Val Reilly, SYBIL project supervisor from the European Commission, for his support and encouragement.

References

Doorenbos, J. and Pruitt, W.O. (1977) Guidelines for predicting crop water requirements. FAO Irrig. Drain. Pap. 24, Food and Agriculture Organization of the United Nations, Rome.

Fedra, K. and Winklebauer, L. (1991) Expert systems for environmental screening. Tech. Rep. RR-91-19, Int. Inst. Appl. Syst. Anal., Laxenburg.

Goldberg, D.E. (1989) Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Mass.

Noell, C.A.W. (1992) Can we put decision support systems into practice? Strategic principles of DSS implementation and integration in agriculture. In: G. Schiefer (Editor), Integrated Systems in Agricultural Informatics. Bonn.

Verhoef, A. and Feddes, R.A. (1991) Preliminary review of revised FAO radiation and temperature methods. Vakgroep Hydrol. Bodemnatuurkd. Hydraul. Rep. 16, Wageningen.

Computers and electronics in agriculture

Submission of articles. Manuscripts should be submitted in triplicate to R.L. Olson (Americas) or to S.W.R. Cox (rest of the world). Review articles should be submitted in triplicate to M.A. Foster. Book reviews to C.J. Studman (addresses on inside front cover). Authors whose native language is not English are strongly advised to have their manuscripts checked by an English-speaking colleague prior to submission.
Authors in Japan please note: Upon request, Elsevier Science Japan will provide authors with a list of people who can check and improve the English of their paper (*before submission*). Please contact our Tokyo office: Elsevier Science Japan, 20-12 Yushima 3-chome, Bunkyo-ku, Tokyo 113; tel. (03)-3833-3821; fax (03)-3836-3064.

All questions arising after acceptance of the manuscript, especially those relating to proofs, should be directed to: Computers and Electronics in Agriculture, Elsevier Science B.V., P.O. Box 1527, 1000 BM Amsterdam, The Netherlands, tel. (+31-20)4853276, fax (+31-20)4853258.

Electronic manuscripts. Electronic manuscripts have the advantage that there is no need for the rekeying of text, thereby avoiding the possibility of introducing errors and resulting in reliable and fast delivery of proofs.
For the initial submission of manuscripts for consideration, hardcopies are sufficient. For the processing of *accepted papers*, electronic versions are preferred. After *final acceptance*, your disk plus two, final and exactly matching printed versions should be submitted together. Double density (DD) or high density (HD) diskettes (3½ or 5¼ inch) are acceptable. It is important that the file saved is in the native format of the wordprocessor program used. Label the disk with the name of the computer and wordprocessing package used, your name, and the name of the file on the disk. Further information may be obtained from the Publisher.

Proofs. One set of proofs will be sent to the author to be checked for printer's errors. In case of two or more authors please indicate to whom the proofs should be sent.

Reprints and page charges. There is no page charge. Fifty reprints of each paper and 100 reprints of each review article published will be supplied free of charge. Additional reprints can be ordered on a reprint order form which is included with the proofs.

COMPUTERS AND ELECTRONICS IN AGRICULTURE HAS NO PAGE CHARGES

All contributions will be carefully refereed for international relevance and quality. Submission of an article is understood to imply that the article is original and unpublished and is not being considered for publication elsewhere.

For a full and complete Guide for Authors, please refer to
Computers and Electronics in Agriculture, Vol. 14, No. 1, pp. 91–97

0168-1699/96/$15.00

∞ The paper used in this publication meets the requirements of ANSI/NISO 239.48-1992 (Permanence of Paper)

PRINTED IN THE NETHERLANDS