

FMS: A SOFTWARE DEVELOPMENT ENVIRONMENT FOR FARM MANAGEMENT SYSTEMS

<http://www.cnptia.embrapa.br/fms.html>

**Silvia M.F.S. Massruhá, Carlos A. A. Meira,
Clevan R. Costa, Renato Fileto**

*Brazilian Agricultural Research Corporation - Embrapa
National Center for Technological Research on Agricultural Informatics -
CNPTIA
Po Box: 6041
13083-970 - Campinas/SP - Brazil*

e-mail: {silvia, carlos, clevan, fileto}@cnptia.embrapa.br

Abstract: This paper describes a software development environment called FMS. This environment aims to generate farm management applications through a semi-automatic process with decreasing cost as time evolves, providing domain experts with tools for rapid and reliable information gathering about their production process. It is also related the current situation of FMS, as well as the experience acquired with this development and its evolution, attempting to be in pace with the current technological evolution.

Keywords: farm management, software environment, applications generator, reuse, specification assistant.

1. Introduction

The Brazilian agriculture, as well as the industry, is being affected by the “invasion” of the computers. The importance of informatics in this area is growing with the economic’s globalization. The same holds true world-wide in general.

The producers need to reach decisions supported by aggregated and quantified information about their production process. In addition, the information about their production have to be reliable to be inserted in the agricultural planning systems. Thus, the computation has an important role to provide information for farm management.

The computer programs for the area of farm management, from the point of view of software engineering (Pressman, 1992), have simple characteristics, with low complexity of the functions and data. However, the importance of this kind of automation is evident for a small farm administrator to take decisions rapidly and to guarantee his place at the agricultural market.

The FMS project aims to establish an semi-automatic software production process for farm management applications, with decreasing cost as time evolves. The production cost of these applications using traditional software development methodology results in an economic First problem. Thus, the need for the automatic generation of these programs. This automatic generation, although simple, requires a large number of tools, methods and techniques and is a current problem at the First software engineering discipline.

The main goal of this research project is the development and integration of methods, techniques and tools to construct a software production environment intended to the farm management domain, providing domain experts with tools for rapid and reliable information gathering about their production process.

The FMS approach is to construct this environment using a platform of programs and tools available at the software market.

2. The Management Model employed

The Farm Management, in the scope of this paper, is divided in two segments: the *management* of the activities that forms the production system and the *planning* of such activities, using information collected through the adoption of the management models (Tung, 90).

The farm management model was conceived to be implemented by the computer, aiming to facilitate the follow-up of this activities. The planning is not being worked on by now for the following reasons (Meira et. al, 1996):

- a good planning is based on a good follow-up;
- the production systems are more robust than the planning models;
- the majority of the Brazilian producers do not have a methodical control of theirs management activities;
- the management automation is simpler and aimed at local application in the farm.

The activities cycle of the proposed farm management model is very simple (Figure 1). The system's user must follow some steps:

- a) the user enters information about the farm and its production process into the computer;
- b) the computer generates schedules of activities in compliance with management practices entered into the system;
- c) the producer goes to pasture and accomplish such activities, trying to follow the scheduled activities reports;
- d) the producer confirms the accomplished activities, entering the data into the computer;
- e) the cycle closes and an other cycle begins with the new activities schedule, and so on.

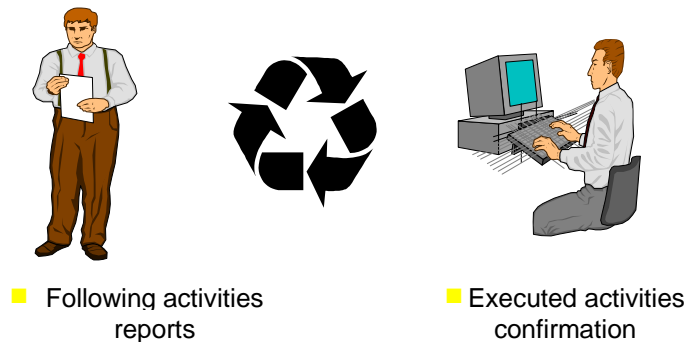


Figure A: Activities accompanying cycle.

For example, consider a Dairy Herd Farm. First, it must be obtained the current situation of the herd. The producer should enter information about the reproductive and productive state of each animal. For example:

Table 1: Cows of a Dairy Herd Farm

Cow	Reproductive State
1	Covering
2	No covering
.	
n	No covering

Using this data, the system generates a report with the activities schedule as described below:

Table 2: Report with the activities schedule

Cow	Activities Schedule	Date
1	Touching exam	yy/mm/dd
2	Insemination	yy/mm/dd
.	.	.
n	Insemination	yy/mm/dd

The producer goes to pasture and accomplish this activities, from this scheduled activities report. Thereafter, the producer confirms the accomplished activities, entering the data into the computer. Then, the new state of the herd and the new activities report are generated. They are illustrated in the table 3 and table 4, respectively:

Table 3: New state of the herd

Cow	Reproductive State
1	Pregnancy
2	Covering
.	.
n	No covering

Table 4: New Report with the activities schedule

Cow	Activities Schedule	Date
1	Parturition	yy/mm/dd
2	Touching exam	yy/mm/dd
.	.	.
n	Insemination	yy/mm/dd.

This is a small example of how the producer can have a detailed activities control of their production system. All activities can be programmed in accordance with the management practices used.

The management practices of an agricultural system depend upon various elements, such as, the region, the animal race and even the culture and the particular characteristics of the farm. This model allows to adjust this practices through specification of some configuration values.

This management model was researched to be applied in various agriculture production systems. Besides agriculture and cattle raising, it can be used in others areas, such as: intern transportation, classification (selection), benefiting (quantity or quality of agricultural products by improvements), packing, warehousing and commercialization.

The management model of a production system can be acquired identifying the management activities and practices adopted. The management activities are called *events*¹. The formal description of each particular production system, in the form of a events model, can be obtained through the methodology described in the next section. This description constitutes the requirements of the application. The requirements transformation process is described in section 4.

3. The FMS approach

The FMS approach is concerned with the specification phase of software development process, where the information about an application are gathered, decomposed and loaded for reuse (Neigboers,1989). These information are decomposed until a representation level where there is an applications generator that decreases the work of the implementation phase.

The Figure 2 summarizes the process used by the FMS project for applications generation. The specification engineer does an interview with a domain specialist, helped by the software environment, to obtain the system requirements. The result of this interview is a documentation containing a functional and data decomposition of the application (Martin, 1985). Such a documentation is represented in the computer using the application language (Leite & Franco, 1990). Then, FMS automatically generates a program for the client's system.

¹ Event is the occurrence of any observable action that causes effect in the system.

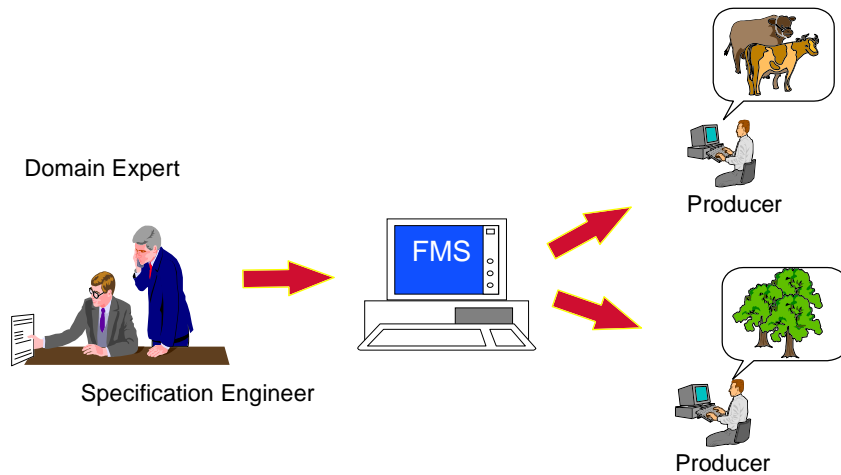


Figure 2: The FMS approach.

The main characteristics and advantages of FMS' approach are described below.

Establish a guide for the specification engineer to interview the domain specialist. The specification phase is organized and aimed to gather the user's requirements decreasing the redundancies and inconsistencies.

Allow that the specification engineer and domain specialist can evaluate the visual aspect and system's behavior during the interview. This procedure is called "Prototyping".

Allow that the interview's information be loaded in a knowledge base so that it can be recovered later to be updated and reused. This decreases the work for the construction of new similar applications. For example, a computer program for Beef Herd Management can use the Dairy Herd Management Specifications or vice-versa.

The role of the specification engineer can be performed by the systems analysts or producers with some knowledge of system analysis and the FMS' environment.

The conceptual aspects and procedures to interview and decompose a production system are described in the FMS' methodology called MEDRA (Applications Quick Development Methodology). This methodology is being studied in parallel with the development of the software environment.

The FMS' approach to construct this software development environment can be divided in four steps:

- determination of similar problems of the farm management domain;
- grouping of these information in two classes: information that can be reused and specific information;
- construction of an applications generator for the first class information;
- specification of a language to gather and load the specific information.

4. The FMS' architecture

The FMS' environment has two main modules to support the approach described (Figure 3): AEsp (Specification Assistant) and GFMS (FMS' Applications Generator).

The AEsp is a program for capturing informal specifications of the farm management applications and converting them, incrementally, into a formal representation (Massruhá, 1996; Massruhá et al, 1994).

The GFMS is a source code generator that automatically constructs farm management applications programs, from the formal specifications provided by the AEsp (Meira, 1991; Masiero & Meira, 1993; Fileto, 1996).

The process to transform the user's requirements into a program is organized in compliance with two hierarchical levels: the conceptual and the operational level. The conceptual level is the view of the user without very much formalism while the operational level has a formal representation.

The operational level is divided in three levels according to the transformation process of the formal specification into an operational program: The application level, the representation level and the implementation level.

In the application level, the system is described using an application language that allows the representation of the functions and data in a knowledge base.

The representation level is defined by the semantic and syntax of the FMS Composition Language (LC-FMS). It is a textual language that is composed by various languages: forms, help, consistencies, apply, events, actions (relational database operations) and reports.

In the implementation level, the system is represented by a program written in the C language.

The translation process of the application level to the representation level generates a LC-FMS specification. The translation of the representation level to the implementation level generates a source code in C language. Thereafter, the source code is compiled and transformed into an operational program.

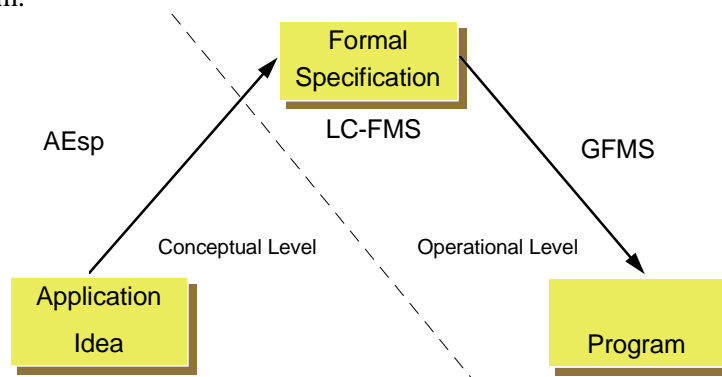


Figure 3 :Requirements transformation process.

5. Conclusions

The FMS' platform is, in its first version, UNIX workstations. The applications generated by the FMS run in PC's under the MS-DOS operating system. This version is composed by an AEsp' prototype and an operational version of the GFMS, but they are not totally integrated yet (Massruhá et al, 1996).

FMS has been used for the definition and generation of various applications, such as a Dairy Herd Management System - LACTUS, a Beef Herd Management System and a System to Manage Vegetal Production.

LACTUS is the first application of the FMS. It is a computer program based farm management system for dairy producers. It currently automates herd and milk production management. The experience acquired in the development of LACTUS enabled its evolution and the FMS' environment evolution too.

The FMS' environment evolution aims to construct a "Front-end" for the GFMS. The textual language of the FMS is being translated into a graphical language. The FMS applications model are being translated into a graphical interface as well.

Other requirement also identified in this experience is reformation of the interview's model. Then, it has been realized an analysis and adaptation of the FMS' Methodology.

A Beef Herd Management System and the System to manage vegetal production are being developed under this new approach.

Acknowledgments

This research was developed under Embrapa's Project (12.0.94.077). The authors gratefully acknowledge contributions by: Mário Ferraretto, Marcos Visoli, Fernando Máximo, Adauto Mancini and Sérgio Passos. We also thank Mário Nascimento for reviewing an earlier version of this paper.

References

- FILETO, R.; MEIRA, C.A.A.; COSTA, C.R.; MASSRUHÁ, S.M.F.S.(1996). The construction of an applications generator using domain analysis concepts (In Portuguese). *In: Brazilian Symposium on Software Engineering, 10, São Carlos. Proceedings.* São Carlos: UFSCar. p.119-135.
- LEITE, J.C.S. do P.& FRANCO, A.P.M. (1990). The Hypertext in the elicitation of the application language (In Portuguese). *In: Brazilian Symposium on Software Engineering, 4., 1990, Águas de São Pedro. Proceedings.* São Paulo: USP/CCS. p.134-149.
- MARTIN, J. (1985). *System design from provably correct constructs.* Englewood Cliffs: Prentice-Hall.
- MASIERO, P.C.& MEIRA, C.A.A. (1993). Development and instantiation of a generic application generator. *The Journal of Systems and Software, v. 23, n.1, p.27-38.*
- MASSRUHÁ, S.M.F.S., FERRARETTO, M.D.; MAXIMO, F.A.; MEIRA, C.A.A.; PASSOS, S.L.Z. & VISOLI, M.C.(1994). AEsp: a specification assistant (In Portuguese). *In: Brazilian Symposium on Software Engineering, 8, Curitiba. Proceedings.* Curitiba: PUC-PR. p.311-324.
- MASSRUHÁ, S.M.F.S. (1996). *AEsp: a specification assistant for farm management systems (In Portuguese).* Campinas: UNICAMP-FEE. 135p. Master Thesis.
- MASSRUHÁ, S.M.F.S.; MANCINI, A.L.; MEIRA, C.A.A.; MAXIMO, F.A.; FILETO, R. & PASSOS, S.L.Z (1996). Software Development environment for farm management systems (In Portuguese). URL: <http://www.agrosoft.com/agroport/docs95/doc10.htm>.
- MEIRA, C.A.A.(1991). *On applications generator (In Portuguese).* São Carlos: USP-ICMSC, 1991. 92p. Master Thesis.
- MEIRA, C.A.A.; MANCINI, A.L.; MAXIMO, F.A.; FILETO, R.; & MASSRUHÁ, S.M.F.S.(1996). Agoinformatics: Quality and Productivity in the agriculture (In Portuguese). *Cadernos de Ciência e Tecnologia, Brasília, v.13,n.2,p.175-194.*
- NEIGHBORS,J.M (1989). Draco: a method for engineering reusable software systems. *Software reusability, concepts and models, v.1, p.295-320.* ACM Press.
- PRESSMAN, R.S (1992). *Software engineering - a practitioner's approach.* 3.ed., McGraw-Hill.
- TUNG, N.H. (1990). *Financial planning and control of agricultural enterprises (In Portuguese).* São Paulo: Edições Universidade-Empresa.