Achim Spangler, Hermann Auernhammer and Markus Demmel, Freising

LBS$_{lib}$ as freely available open source model

With LBS an electronic communication system for mobile agricultural machinery is available in Germany and worldwide for the first time. Data transference in practice often shows
Compatibility problems caused by differing layouts of the standard or individual interpretation and extension. These difficulties can be solved with the LBS library LBSlib which is freely usable for anyone as open source model. First applications have confirmed the simple utilisation and reliable operation of the developed system.

Dipl.-Inform. Achim Spangler is a member of the scientific staff in the special department Technology in Plant Production in the special sector for Biological Raw Materials and Technology in Land Utilisation of the TU Munich. Prof. Dr. Hermann Auernhammer is manager, and Dr. Markus Demmel assistant, in the same special department, Am Staudengarten 2, D-85354 Freising;
e-mail: *spangler@tec.agrar.tu-muenchen.de.*

Keywords

LBS, CAN, Open Source

Support information

The part-project „Process data recording in the agricultural BUS System (LBS)" is supported by the German Research Society (DFG) within the research group „Information systems for small-area crop management IKB Dürnast".

Literature details are available from the publishers under LT 01315eor via Internet at http://www.landwirtschaftsverlag.com/landtech/local/fliteratur.htm

Electronic and electrical communications have in the meantime become irreplaceable parts of tractors and farm implements. A real pioneer performance was achieved in Germany from the middle of the 80s for unlimited use on farms through standardisation work for the Agricultural BUS System (LBS) according to DIN 9684. This standard serves in the meantime as basis for an international standard being worked on at the moment (ISO 11 783) with which the electronics in all tractors and implements should be equipped in future.

But despite the available standard and transformation for the agricultural engineering industry (the first tractor manufacturers offer LBS as standard equipment), the utilisation in farming has only sporadically progressed. An important reason for this is the fact that despite standardisation, compatibility difficulties repeatedly occur in practice. There are several reasons for this:

- as a written standardisation, LBS is very involved and complex with more than 130 pages
- not all possible situations are clearly and unmistakably established, which leads to manufacturers translating their own requirements according to judgement or experience
- parts of the standardisation can be interpreted in different ways or even interpreted wrongly
- there is as yet no neutral testing station for checking and evaluating standardisation conformity

A rapid and mistake-free transformation (and this applies in the same way for ISO) can therefore only be achieved when the compatibility problems can be solved.

Solution step

Compatibility problems can very easily dealt with when the total interpretation and coding of the required communications software lies in a single hand and is carried out by the same organisation, too. Unanswered questions in practical transformation and unavoidable mistakes thus involved, must be processed in open discussions between developer and user working closely together and then brought into the basic software.

The philosophy of „Open Source Models" – comparable to „LINUX" – was created for LBS and in the meantime transformed into LBS$_{lib}$ (*fig. 1*).

Subsequently, the unique coding station in Weihenstephan was created. Early-on the potential user and the possible testing organisations were involved so that a steady inflow could be experienced of experiences, successes and remaining problems.

According to the Open Source Idea, all part-takers undertook, in agreement with the „Lesser General Public License" (LGPL-License):

- to use the software only according to licensing agreement
- to make available free, and immediately, to all other users alterations (e.g. corrections, optimisations and extensions)
- to include information on commercial products used with Open Source Software or at least to inform where these can be got

Guaranteed through this requirement to declare all alterations is that all improvements flow into a reference version of the software. In this was it is obvious to all which product the program library is serving.

An internet platform was created for communication between exchange and user. There, the preparation took place of:

- the library in source code with comprehensive documentation
- simple utilisation examples for starting off
- extensive examples for a rapid, practical transformation

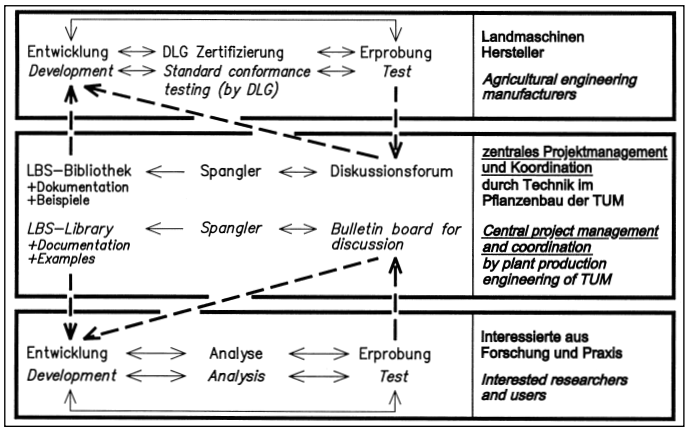

Fig. 1: Structure for the development of LBS$_{lib}$

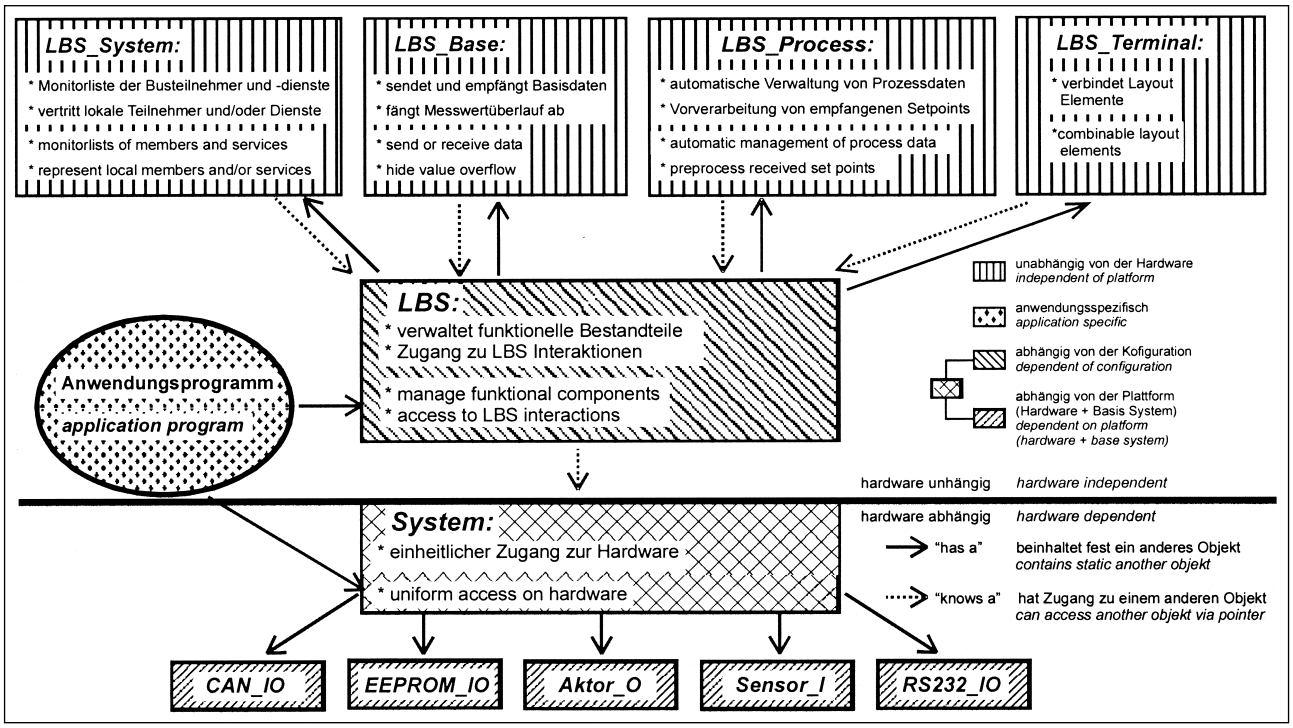

Fig 2: Structure of the library of LBS$_{lib}$

- a discussion forum for open questions, emerging problems, alteration suggestions and for the information of those starting-off
- tools for the library administration, right of access and information distribution

Design and structure of LBS$_{lib}$

The design of the library covers the total LBS standard with the necessary objects. The communication conforming to LBS and the targeting of the hardware is split into two parts (*fig. 2*).

The first comprises:

- LBS-system: administration of those taking part and those servicing
- LBS-base: basic information from tractor and terminal
- LBS-process: suitable communication with the process parameters
- LBS-terminal: targeting of virtual terminal (optional)

A possible extension could, at this point, involve the loading of screen masks according to LBS+.

Within hardware targeting, at least the object to the CAN communication (CAN_IQ) must be involved. Drivers for the sensors, tractors, serial interfaces, EEPROM and PCMCIA memory-flash-disc can be optionally activated.

A limiting of the hardware-dependent elements substantially reduces the effort required for matching the new hardware platforms. With this in fact, a very large proportion of the respective applications be taken over unchanged.

The software was developed as object-oriented. Through this, the whole system is built on single modules whereby every element encapsulates a partial task which can be separately developed, tested and optimised.

This building block principle is supported with program languages such as C++ and JAVA. This guarantees:

- simple descriptions of the interactions within the BUS system
- security in real time applications
- servicing and extension possibilities
- flexibility in the conducting of tasks

Transformation and availability

In the meantime the extremely complicated work of the system definition, implementing and accompanying tests of the LBS are almost completely finished. Subsequently, transformation and usage follows in two parallel projects.

LBS$_{lib}$ workshops

The processed library with practical examples has in the meantime been presented to interested users in three national and international workshops.

Practical applications for the automatic process data recording

At the same time a transformation took place within the research group IKB Dürnast (see *http://ikb.weihenstephan.de*) for the automated process data recording. For this, during work with a grubber equipped with an implement identifier (IMI) was recorded with geo-references draught power, fuel consumption, wheelslip, driving speed and other working parameters. The total programming took place with the required objects from the LBS$_{lib}$.

The system equipped in this way demonstrated its functional suitability in 100 hour operation. The evaluation software IMI$_{lyzer}$ which had been developed in the meantime enabled a first wide-reaching automatic analysis of the most important process parameters from the recorded data (*table 1*).

In spring of this year the total tractor, machinery and implement fleet of the Dürnast experimental station was equipped with such systems. Thus, the long-term test and further improvements of the library module took place at the beginning of the vegetation period in a comprehensive field trial.

Table 1: Process data „Stubble tillage Schafhof 2000" (Fendt Favorit 714, Lemken Smaragd 3 m working width, 14. 8. 2000)

Operation start	12:25	Working time	4.64 h
Operation end	18:48	Turning/empty	1.11 h
Total time	6 h 23 Min.	Stopped time	0.63 h
Area worked	12.24 ha	Prop. work time	73 %
Total distance	49.85 km	Prop. turning time	17 %
Working distance (wd)	40.79 km	Stopped time	10 %
Proportion of wd	82 %	Prop. DGPS-	99 %
Area performance	1.92 ha/h	Prop. GPS-	1 %
Work speed	8,82 km/h	Average draught requirement	7684 N
Average specific implement resistance (at 3 m ww and 15 cm wd)			171 N/dm^2