All 12 railway administrations in Europe have already experimented with Computer-Based Interlocking Systems (CBIS) and are now installing them in preference to Relay Interlocking Systems (RIS), with the cost benefit being the main justification. In total, 142 CBIS are in service in 10 out of 12 European countries (until the end of 1995). With the systems commissioned during the last few years and including the urban transit (metro) and private railways this number exceeds several hundreds.

The standardization of the safety and functional requirements for the CBIS is under consideration in Europe. The CBIS promises standardization by nature of realization of interlocking functions by software instead of hardware (relay contacts), but there is still a disagreement between the systems developed and applied in different countries. After the analysis of the existing European CBIS the following has been concluded: “Considerable efforts should be made to ensure that the software - a major investment - can be recovered to be run on different hardware, subject to extensive development; if not, the economic benefits of data processing will be greatly reduced”.

The purpose of this book is to contribute to the standardization of CBIS. The most suitable hardware, proven by practice, is selected and some improvements are proposed. Software of the existing CBIS is hardware dependent and represents the interlocking logic of a particular railway authority. These logics are different as a consequence of a country specific signalling principles and practice of implementation. In addition, the existing unchangeable portion of interlocking software is very small, while data preparation is extensive and time consuming. The main contribution of this book is the development and implementation of general interlocking software, which is independent of hardware and country of application. In

addition, the unchangeable portion of interlocking software is maximized, while the data preparation is replaced by the production of a simple standardized data file.

During the last two decades of research in the transportation systems field many types of CBIS have been developed, implemented and approved worldwide. During the application phase many disadvantages have been observed. Some are: small application can be more expensive than an equivalent RIS consisting of 100 to 500 relays; high cost of the hardware specially designed to suit safety requirements; superfluous hardware in small or non-standard applications; insufficient computer resources (speed, address, memory); limited number of trackside elements; eight-bit processor limitation (Motorola 6802 or Intel 8085); use of specially designed software for data preparation and programming; slow operation due to the serial nature of processing; significant number of safety functions application dependent; automatic route setting features usually excluded from the safety system level; indications on the screen restricted, especially for flashing aspects; not flexible enough to accommodate the range of trackside equipment found in different countries; not easily changeable to suit specific railway demands; a considerable amount of total time required for: design, checking, testing and commissioning; an expensive senior signalling engineer required through almost all stages of realization; a considerable amount of time required for simple alterations; maintenance messages typically coded and as a result not directly understandable and operator interfaces not intuitive and requiring specialized training of all users.

This book summarizes a means of improving existing CBIS. The result is presented as Universal CBIS using a standard data structure, independent of track layout and country of application. The main improvement is a direct conversion of the operational, functional and safety requirements of an

interlocking system into general interlocking software. The approach is based on a novel standardized method of presenting traditional control table data such that it can be entered as a simple data file suitable for computer application.

Universal CBIS has been designed from a signalling engineer's point of view using, as far as possible, commercially available computer technology and experience gained from the application of existing CBIS. The system is a modular, distributed, fail-operational, either centralized or decentralized computer system.

Universal CBIS consists of: PC, keyboard, VDUs, printer, central processing unit (CPU), recording media, remote processing units (RPU), trackside equipment with appropriate interfaces (signals, points, track circuits etc.) and a power supply. The system can be remotely controlled as part of a Central Traffic Control (CTC) system and can allow remote maintenance access as part of a Central Maintenance System (CMS).

The existing CBIS are based on various fail-safe techniques and concepts. From the safety point of view the two-out-of-two system (duplication for safety) is a fail-safe system equivalent to the RIS, but a two-out-of-three system provides much better availability characteristics. Therefore many approved systems utilize this concept and it seems that such a configuration has become a standard. The high price of specially designed processor modules was the main reason for developers to adopt solutions with lower numbers of processor modules. We believe this is not an issue any more. Due to the significant reduction in the price of computer equipment, generally, redundancy of hardware, for the purpose of achieving safety and availability, appears to be a cost-effective solution. The CPU of Universal CBIS consists of three identical processor modules, which operate, in a triple redundant fault-tolerant configuration with redundancy management.

The redundancy management hardware is proposed as a simple and reliable unit, independent of the processor modules. Hence, the cost of hardware will be minimal while the decrease in its complexity will contribute to its higher reliability. The comparison by software will be quick enough due to use of high-speed processor modules.

RPU is an intelligent dual fail-safe interface controller acting as a concentrator for a group of trackside elements. Data collected by the unit can be stored in local RAM and transmitted later at one time. The main purpose is to allow the connection of remote equipment without expensive signalling multicore cables and to improve the speed of reading the element's states from the field. Low cost industrial controllers can be used for the realization of the unit, due to limited local functional requirements. RPU can also satisfy various temperature requirements of the environment. Communication is based on vital serial transmission with protective coding and the application of a Cyclic Redundant Code (CRC) check-sum. Communication uses a master/slave protocol wherein the units talk only if questioned by the CPU. The existing CBIS have a communication speed, which is generally low, with little possibility of being increased using current hardware. The proposed solution allows an increase of the speed up to the highest speed of available industrial solutions at the time of realization. Direct Memory Access (DMA) and an interrupt handler could be used to increase the speed, where required. In the case of failure the unit will be isolated and graceful degradation of the system will take place. Careful arrangement of the equipment controlled by the RPU can minimize the effect of graceful degradation on availability of the whole system.

A general type of the interface is based on a simple, commercially available, standardized microprocessor interface. The system deals with the trackside equipment as controlled objects, through input and output registers,

making the hardware independent of the control system requirements. This means that various equipment found in different countries can be connected to the system by provision of appropriate low cost interfaces. Thus the proposed system can be utilized for various countries without significant alterations to the hardware and almost no alteration to the software. The interface is not fail safe, but its function can be proved in a closed loop of real-time control before and after the change of the state of any element. The application of memory mapped Input/Output (I/O) space and vector interrupts can be used in order to increase the number of outside equipment and maximize the performances of the system. The basic fail-safe realization of the circuits for the trackside elements (signals, points and track circuits) is retained.

Special hardware is proposed for comparators and redundancy management hardware, rather than for redundancy modules, to allow hardware compatibility and easy safety proving. The redundancy management hardware does not have to be fail-safe as a unit. It can be checked by software using specially designed techniques, which will prove the correct working state just before the voting process, as well as immediately after the voting process, giving the opportunity to cancel the previous decision if a failure is detected. Fail-safe redundancy management hardware is still desirable due to the difficulties in proving the software safety. It also releases software from permanent checking and improves the system speed.

The comparison of the outputs is not necessary for each processing cycle. It is satisfactory to prove the agreement of the modules only when safety critical activity is demanded. This will happen whenever the system sends a command to an element to change its state. This will eliminate clock synchronization problems as well as superfluous crosschecking between the

processor modules. The reduction of the comparison frequency allows a simplification of the redundancy management hardware and/or the use of faster processors. As a result the features of the system can be improved without any loss in safety or functionality.

The most important safety approach is based on the classical and widespread safety analysis of RIS. The main characteristics of the safety analysis are single channel information flow and a system resistance to a single fault. This means that a single failure cannot cause unsafe conditions under any circumstances. The calculated safety for a single CBIS, expressed as Mean Time Between Unsafe Failure (MTBUF), required by the Office for Research and Experiments (ORE) of the International Union of Railways (UIC) is 100 years. The reliability required by ORE UIC is 4 months.

Universal CBIS offers the same treatment of a CBIS as that of a RIS from the safety point of view. Unpredictability of the failure modes of the electronic components is covered by isolating the faulty module, giving the system at least the same level of safety as that of an equivalent RIS. The calculated safety, even for a two-out-of-two RPU, satisfies ORE UIC requirements. The triple redundancy technique, used for realization of the CPU, improves reliability and availability of the system. Modular structure makes replacement of the faulty module fast and easy, contributing to the higher availability of the system. The probability of the same error occurring simultaneously in two different processor modules is very small. In addition, diversity in hardware and software can be used as a protection against common mode failures. There are solutions with two times two-out-of-three systems and two different sets of software. Theoretically it is possible to reduce the probability further by increasing the number of elements, which must be in agreement, three-out-of-four or four-out-of-five, but the cost of the system will be higher and could not be justified. The mathematical

analysis of the triple redundant CPU, with repair cycles, indicates that expected failure rates for the system and for the parts satisfies ORE UIC requirements.

Universal CBIS is a low cost solution based on commercially available microcomputer hardware. Instead of using specially designed hardware, railways can now benefit from general industrial development. High performance processor modules with huge memory space, together with a higher speed of data transmission will significantly improve the speed and capability of the system. The limitation present in most of the existing CBIS, regarding the number of controlled elements, will be almost completely avoided. The system is flexible and upwardly compatible without the disadvantages related to the maintenance due to fast advances in technology. This is achieved by using a high level programming language instead of an assembler language dependent on a particular processor. As a result software with advanced capabilities can be used, too. Hardware and software can follow the advances of technology and the cost advantages will become progressively more pronounced.

The most suitable programming languages (highly recommended by CENELEC standards) for integrity levels 3 and 4 are safe subsets of the following languages: Ada, Pascal, Modula-2 and Fortran 77. Ada and Pascal are widely used in railway signalling industry. The development tools and validated compilers are available for both of them. The latest developments of Ada, which improve visibility of the object code and simplify the task of validation, may favour the use of Ada in the future.

The maintenance of Universal CBIS is improved by the use of advanced diagnostic and self-diagnostic software. The maintenance terminal is usually a standard part of the system. It is based on a commercial hardware. Instead of employing distributed maintenance terminals, a CMT is proposed. This

reduces the demand for maintenance personnel and provides advanced maintenance features. A remote fault detection and location can be immediate and maintenance response can be faster and more efficient.

In RIS and some CBIS today safety related auxiliary control actions are recorded by means of counters and additional manual written notice. A “black box” for recording of safety critical actions is introduced for Universal CBIS. Thus, registration of an irregular situations and its traceability can be performed and used for post-incident analysis. The responsibilities for an accident can be established as well. Therefore, the psychological effect on the operator will be positive: to improve his/her attention. This will have a preventive effect.

The use of commercial software is likely to feature strongly in future safety-related systems. This software can be used for the operating system, standard protocol or a database system. In this book special attention is paid to the interlocking software. The interlocking software is divided broadly into two groups: application driven data and general interlocking software. Application driven data are used to define the layout of the station and to create the computer control table. The most important part is general interlocking software independent of the application.

Operational, functional, and most importantly the safety requirements are listed in the control table. The translation of control tables into a form acceptable for use by the computer system is a difficult and time-consuming task requiring limited computer knowledge. It is preferable to retain a traditional form of the control table for CBIS application. A specific design language has been developed to describe the requirements of particular installations in which data preparation and translation are made computer readable. The control table proposed for Universal CBIS represents an efficient way of data preparation and results in the formation of a simple

CBIS database. Although individual signalling engineers have a freedom to define particular requirements, the fail will be computer readable control table.

The interlocking safety principles, rules and regulations are very complex and particular. The most important safety principle defining interlocking between routes requires that no two routes share any portion of the track at the same time. If two routes have a shared portion they are conflicting routes and cannot be set at the same time. The proposed control table is based on this principle and all other applicable safety principles. Some parts of the control table, such as approach time locking, time release, aspects and aspect sequencing, etc., are omitted from the control table and inbuilt into the general interlocking software to allow generalization and simplify data design.

A user-friendly program has been developed to make the input of the control table related data such as: number of routes, number of elements, values of the timers etc. in an interactive fashion with confirmation. The control table itself will be entered into the software in the form of interlocking functions expressed in Boolean form. This method has been chosen to allow simple and easy input of data, as well as to simplify checking and Quality Assurance (QA) process. As entered, version of the data will be produced by the program to allow immediate checking and corrections until a version free of errors is obtained.

A user-friendly program has, also, been developed to make the design of the screen layout as easy as putting the predefined indication element modules (“mosaic fields”) together to form the appropriate picture. All the required elements have been developed and they are available in an elements library. The elements are placed by specifying coordinates to suit the signalling arrangement. The actual indication, as in the field, including

flashing aspects, will be available to the signalman all the time. This is an improvement to the existing CBIS and RIS. Generally the systems do not differentiate between the proceed aspects, and only one proceed indication is given. This could, for example, disable a signalman from undertaking preventive action in the case of failure. In some systems flashing aspects are indicated by adding a flashing letter next to the aspect. The non-appearance of the letter, which is caused by an error in the indication part of the system, could confuse a signalman.

CBIS are designed to be generically applicable. This is normally achieved through the use of a common set of software (interlocking software), which is configured using data specific to the location. Data are often produced using a defined meta-language, and a data development lifecycle can be identified which is very similar to the software development lifecycle. The typical size of common interlocking software is usually very small, while data occupy much more of memory space. SSI system (Alstom), for example, has 12 Kbytes of common interlocking software and up to 40 Kbytes of data. The basic idea applied for Universal CBIS is to use, as far as possible, knowledge and experience of the signalling engineer and implement that into the software of the system. Therefore the proposed general interlocking software is significantly larger than common interlocking software of the existing CBIS. In this way, the system incorporates expert knowledge, basic signalling principles, common railway rules and regulations, and replaces the signalling engineer through all phases of the design of an interlocking system. The size of the proposed data, representing a particular station, is significantly reduced, when compared to the data of the existing systems. For a small station the size of all data files is less than 3 Kbytes.

The software of the system is designed in the form of a main program and various functionally oriented sub-programs, representing the synthesis of safety as well as functional and logical requirements based on the single-channel operating principle. The general-purpose interlocking program embodies most of the standard signalling principles and contains the rules for operating on the control table data to produce the precise signalling controls required. In addition, most of the algorithms for performing interlocking functions were specified as far as possible according to those of the conventional RIS, which has been proved to be valid through half a century's experience. This enables easier safety, functional and logical analysis of the software.

Railway Signalling Principles: Railway signalling knowledge and experience are expressed through safety railway signalling principles. The general principles are valid for all systems regardless of the country of the application of an interlocking system. There are some differences between different country's systems, but they are more functional and operational than safety in nature. The principles are complex and unique to the railway signalling field, but it is important that they can be generalized and converted into software with the aim of developing a general CBIS. Signalling principles and safety functions are completely built into the general interlocking software. The designer is released from the task. By implementation of the control table and track layout all jobs are done. The program itself puts all the parts together and makes all the required interlockings.

Route Approach: There are, generally, two well-known ways of designing signalling circuits: "spoorplan" (modular geographical) and "free-wired". The "spoorplan" system has in-built redundancy contributing to generality and making the design easier, it but increases the cost of hardware. The

"free-wired" system is tailored to the requirements and consequently hardware is less expensive, but the design is more expensive due to the need to solve the problem from a case to a case. It is important to emphasize that there is no basic difference between the system's functions. The pattern is a chain of self-contained circuits operating in cascade, each performing a function and passing the result to the next in the chain. A route approach is utilized in the software. The route concept combines both the general "geographical" approach and the efficient "free wired" approach. It allows generalization without spare capacity in either hardware or software. Direct correlation is established between the interlocking function and the route. Interlocking functions and subfunctions are written very simply, but they are used extensively in many different ways in various software modules. Software modules are developed to cover all "chains". The program flow follows the approved cascade operation of relay circuits.

Automatic Route Setting (ARS) feature releases the signalman from unnecessary presetting of the elements required by the route and gives him/her time for other activities. A remote control of an interlocking is simplified due to the use of one route command instead of a set of unit lever commands for the elements and a command for the route itself. The ARS feature is achieved in the general interlocking software rather than by panel processor software as in most of the existing systems. This approach gives functional consistency on the route level inside the interlocking as well as individual manipulation of the element if required.

Universal CBIS adopts a modular structure of hardware and software. The system can be built to the required capacity from modules and configured to suit the application. Thus, only the amount of hardware actually required for an installation needs to be supplied. Modular design makes the applications for various countries more adaptable. The country

specific issues will require alterations of only a few modules. This makes software checking, testing and validation easier and faster.

The simulation and testing in laboratory conditions is simplified. There is no requirement for a special workstation and bulky manuals. All activities can be done on the system itself or any PC. The simulation software is the actual general interlocking software using simulated inputs. The trackside equipment can be tested with the system, or separately by the simulation of outputs and checking the responses. The simulation software can be used as a powerful independent checking tool for checking the conventional control table design. This will enable the uncovering of problems before installation and save time and money, especially in the case of conventional RIS, where changes affect hardware. The simulation software can be a very useful training tool for designers and checkers, giving them the possibility to correct their mistakes and develop their skills.

The latest CANELEC standards, covering safety-related and safety-critical software, have introduced the concepts of safety integrity levels. The proposed general interlocking software belongs to the highest safety integrity level “4”. VDU-based signalling control systems have now gained widespread acceptance and current trends suggest that the functionality available within such systems is increased. VDU-based software is now, in accordance with the safety standards, recognized as safety-related software (levels “3” to “1”). The proposed VDU software belongs to level “3”. Following the standards does not guarantee that the resulting system will be fail-safe, but it allows the fail-safety of the system to be assessed. All software modules have been rigorously tested in accordance with the standard principles of safety analysis of RIS. The general interlocking software is tested with the data presented in the computer control table and

with software, which simulates inputs from the field. The results were satisfactory.

Verification and validation of the safety critical software are required. This task has to be done in accordance with the latest standards. Current best practice for any large computer based system would include nearly all of the phases recognized by the safety standards, other than the safety specific activities such as hazard analysis and risk assessment. That task has to be performed in the form of safety analysis of the general interlocking software in accordance with the safety analysis of RIS. This proving should be done by experienced signalling engineers. Validation of the software modules, representing special requirements of one country, can be done by signalling engineers from that country to allow the best translation into the software. Once general interlocking software has been approved, design becomes easy and simple. Instead of the development of advanced techniques to make data preparation and checking easy, design, checking and testing process is significantly reduced.

Functional testing of the existing CBIS is carried out in the office on the design workstation. The time required for testing varies from approximately one week for small stations to several weeks for larger and more complicated stations. Functional testing of data is mandatory for all integrity levels. This simply recognizes the fact that data production for control systems is not very different to writing software. Universal CBIS obviously does not require any programming for data production, so this part of the testing can be omitted.

Formal proving methods of software have not yet gained wide acceptance, and for the existing safety-related systems it seems unlikely that they will, at least in the short term. However, Universal CBIS gives the opportunity for the application of formal methods. For all stations, which

comply with the initially selected type of station, it can be proved that the general interlocking software will always generate correct functions. For stations with some slightly different characteristics and which, therefore, do not completely belong to the initially selected type of stations, limited functional testing will be required. Consequently, factory acceptance testing (FAT) can be either completely omitted or significantly reduced.

The hardware and software modules used in the system are flexible and can be configured to suit the most diverse of customer requirements. The control table exactly determines the size of the software and hardware of the system. There is no need to provide superfluous hardware and software modules for any interlocking system. As a result the size of the proposed CBIS is optimised for each application.

The most significant advantage of Universal CBIS is much greater implementation of railway interlocking safety principles and general knowledge and experience of the signalling engineer into the software of the system. This approach makes the system design more independent of the signalling knowledge and reduces the need for the presence of signalling experts who are not easily available. The reason for the shortage of signalling engineers is that they are getting their skills only through work experience since there was no opportunity to get them educated by other means until recently. On the other hand, for data preparation of existing CBIS, computer knowledge is very important. This excludes a significant number of experienced signalling engineers who can easily handle relay circuits. The proposed solution overcomes this discrepancy by dividing the job at a natural border. A valuable experienced signalling engineer can continue to deal with the principles and safety requirements including validation of the system. The signalling arrangement and control table have to be produced by a signalling engineer. The signalling arrangement will be

the basis for the design of the station layout for the VDU. The control table will be used as a knowledge base for the system. All other stages of the design and the development of the system, until the final testing, can be undertaken by programmers. For the final testing and commissioning, the presence of a signalling expert is required again. The final test will prove all safety locking and functional requirements against the control table and point out alterations to be made before complete satisfaction with the design requirements is achieved. This process is very similar to checking of existing RIS. Hence, a checker does not have to be a computer expert, signalling knowledge alone is sufficient. Considering the very short time required for design and checking activities, all required corrections can be within a day. This reduces unexpected delays to the commissioning, minimizes risk and allows better planning of the project.

The programmer will be asked by the program to input all required data. Through this process all geographical and other relevant data will be entered. The time for both design of the station layout and for entering the control table, even for very complicated configurations, is significantly reduced. The computer will then print the entered control table data. The computer control table will have the same format and layout as the originally designed one. This will make checking as easy as a simple comparison of two numbers. The appropriate corrections can be made straightaway. The time required for this process is dependent on the size of the control table, but in most cases it will not require more than a few hours. The paper work, design and checking documentation are also reduced to a minimum. The same procedure is applicable for further alteration of the control table caused by the alterations of the layout after the commissioning. This way is very easy, fast and less expensive than the alteration of an existing CBIS.

Universal CBIS with commercially available computer hardware and general interlocking software written in high-level structured programming language independent of the hardware has been proposed. The safety of the system is based on a proven practice of triple redundancy with pre-defined repair time and appropriate safety techniques. Trackside equipment interfaces are developed as the interfaces to the real-time controlled objects and generalized to suit various types of equipment used by different countries. The system is upwardly compatible and open for further development and alteration to satisfy the most diverse railway requirements. Both hardware and software are designed on a functional module basis to allow the advantages of the modern concept of the fault-tolerant real-time controlled systems to be applied. The most important railway signalling knowledge and experience is implemented into the software of the system making it general and independent of the layout of a railway station. A simple and easy method of defining the control table and VDU layout design is proposed to make the development of the system very quick, cutting the cost of the whole project. The design, checking, testing and commissioning time is reduced to a few days from over a month.

Further work is required to perform a validation and a verification of the proposed safety critical software. The selection of the most efficient redundancy management software, diagnostic and self-diagnostic software will also be required.