Analytical Laboratory

Controlling of Analytical Instruments and Laboratory Management

The Wiesbaden Computer-integrated Laboratory is developing laboratory integration software

Chemical laboratories have no alternative but to introduce higher degrees of automation thanks to growing complexity of processes, fewer personnel resources, and an increasing number of assays. In addition to this, in the food and pharmaceutical industries, the requirements of the authorities are becoming stricter and stricter, particularly with regard to software applications which support automated processes. At the “Wiesbaden Computer-integrated Laboratory (WICIL)” in the Computer Science Department at Fachhochschule Wiesbaden - University of Applied Sciences, Germany, a group of professors and their students, headed by Professor Reinhold Schäfer, have been developing automation software for 11 years now. This project is being sponsored by research grants from the State of Hessia and the industrial sector. The sub-projects involved – usually dissertations and special projects – are being conducted in collaboration with renowned research institutes and approx. 20 companies in the industrial sector. Worthy of special mention are the National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, the US National Laboratories in Oak Ridge, TN (ORNL), and Los Alamos, NM (LANL), as well as Hewlett Packard Co., Agilent GmbH, and Creon-Labcontrol AG.

Reinhold Schäfer

Keywords

Integration, laboratory automation, LIMS, graphic programming, device plug-and-play

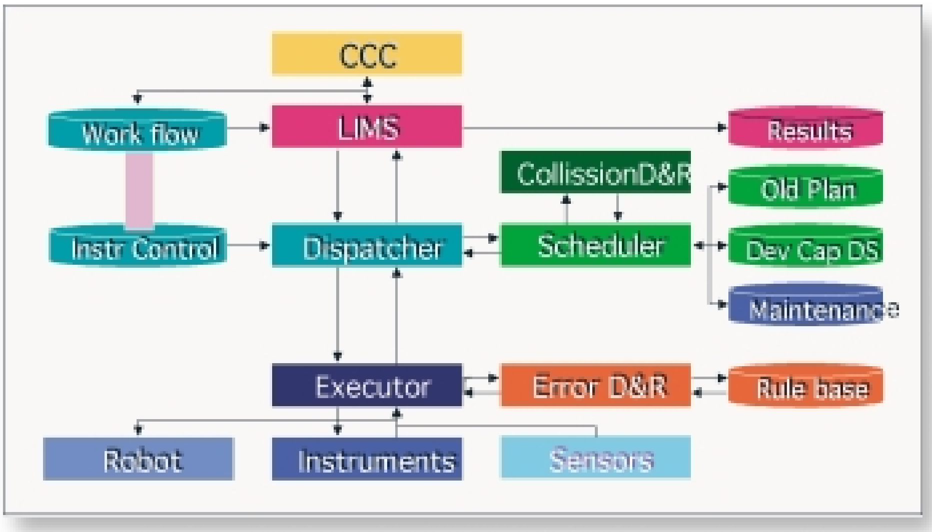

Fig. 1: WICIL system architecture

Overview of project

The aim of the project is to enable uninterrupted, 24-7 operation using a consistent system architecture (Fig. 1) [1] [4] [10] [16]. Any laboratory procedures can be created using a graphic editor and processed automatically after assignment of the samples for determination. For this purpose, the processes can be described with any degree of granularity, ranging from the execution of an entire "content uniformity" test right down to the tiniest detail such as the opening of the door of a set of scales or reactions to error states. A dynamic scheduler optimises processing of samples on the laboratory bench. To do this, it determines the optimum order for the processing of the various activities, so that as many samples as possible can be processed in parallel without impairing the progress of each other. In addition, ad hoc analyses, reports and disturbances on the laboratory bench are taken into account by means of dynamic rescheduling [2] [9]. The plan, which is set well in advance, can include potential collisions between devices on the laboratory bench. These, however, already show up in a collision recognition module during the planning phase and are used in the correction of the final workflow adopted. As a final step, a laboratory console is used for monitoring and simulation, if appropriate with generic devices. Reports and errors are processed using a rule-based event and exception handling module with graphic rule definition, fact conditioning, and simulation.

Graphic creation of workflows with REGULUS

In the laboratory, the purpose of workflows is to provide laboratory technicians with a set of instructions describing the exact order of a series of steps in a process. If special circumstances occur, alternative working steps or the ad hoc checking of certain conditions must be possible. In the case of an automated laboratory, the workflows are also composed of device-related activities including controlling of the functioning of devices and the transfer of results. Often only manual processes are transferred to automatic machines (robots, capping stations etc.). Workflows are, however, increasingly being adapted to suit specific devices or completely rearranged. It is important to incorporate sensors which detect the changes in status within a workflow on the laboratory bench or establish the location and status of samples, so that these can be evaluated and the subsequent steps selected or configured. Also, data have to be processed and stored. Queries, loops and conditional branches must be available as operators. The system must also incorporate concurrent operators in order to optimise the automation of the order of elements in a workflow.

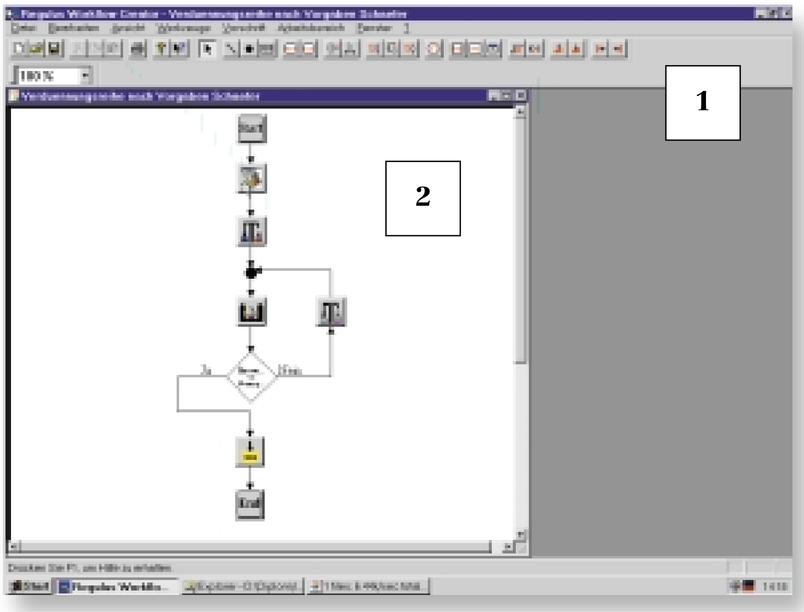

Fig. 2: REGULUS screen (1: Toolbar, 2: Workflow graph)

An extended workflow diagram is used as the paradigm for workflow production (Fig. 2) [13] [15]. Rectangles carrying symbols (service units) stand for devices, sensors, actors, formula processors, storage operators and sourcing and sinking of samples, materials and containers. These are connected by arrows which represent temporal sequences. Variables can be defined and processed in service units. During the compilation of workflows, the user can ask for specific parameters via special dialogs in the REGULUS software package (e.g. instrument preparation times, duration of procedures, maximum delays before subsequent steps, duration and number of interruptions in an activity). Entire workflows can be grouped into macros, parameter transfer protocols attached and stored in libraries for reuse. Finally, the range of REGULUS functionalities is rounded off by a number of different configuration dialogs (Figs. 3 and 4) [16].

Optimised scheduling and re-scheduling

In addition to the working processes themselves, the control parameters for each activity are also documented in the procedure management. Amongst other things, this includes their preparation time, duration (Fig. 3) and the time constraints for the subsequent activity [2] [9] [13].

Depending on the type of optimisation preset, the task of the scheduling software is now to distribute each activity across the resources needed, so that either the throughput of the analyses is maximised or the individual samples are processed as rapidly as possible [3] [11].

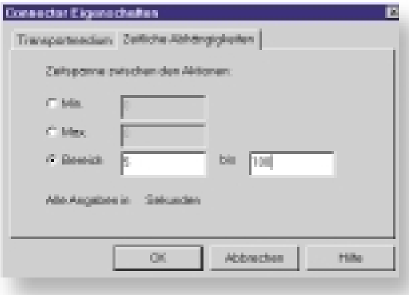

Fig. 3: Connector property definition (minimum and maximum delay of activities)

When scheduling, a range of chemical constraints must be taken into consideration. For example, activities with well-defined durations in particular must not be interrupted, since the results of the analysis would otherwise be falsified or the samples might even become unusable.

Significant time savings can be achieved by exploiting the properties of multi-position instruments. In this way, it is possible, for example, to process several samples on one shaker under the same shaking conditions, even if the processing time required is different. The shaker only has to be programmed to stop at preset times and the samples taken successively from the shaker (Fig. 4). This property can, however, only be used if the user explicitly permits this when loading the samples.

If any errors occur on the laboratory bench – e.g. if an instrument fails – or if an ad hoc sample is inserted onto the workflow, the scheduler calculates a new optimal plan taking into account the samples still being processed and the processing time for the plan. In doing so, of course, it must be ensured on the one hand that the processing of samples already in the device must be able to continue as smoothly as possible, and on the other, that the maximum times set in the workflow are adhered to and that non-interruptible sequences are not disturbed.

The planning data of the scheduler can be checked in the form of a Gantt chart (Fig. 5). Calculated plans are passed on as a standard procedure to the executor, which runs them in a device-specific manner in the distributed system supported by CORBA [3].

The scheduler accesses the description of the devices, their geometry and dependencies, as well as further resources in the system capability dataset (SCD) [6] [7] [8] [11]. In addition, it has to plan for necessary preventative mainte-

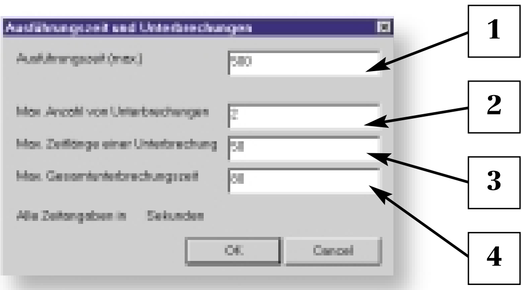

Fig. 4: Requirements for multi-resource instruments (1: activity duration, 2: maximum number of interrupts, 3: maximum duration of one interrupt, 4: total duration of all interrupts)

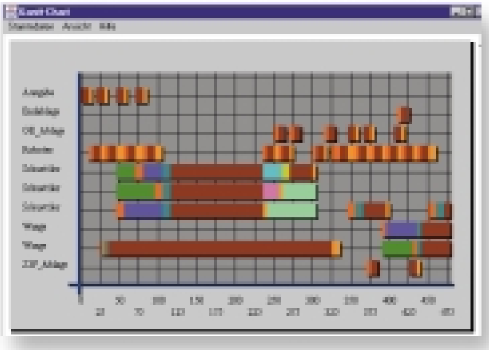

Fig. 5: Gantt chart (different resources over time)

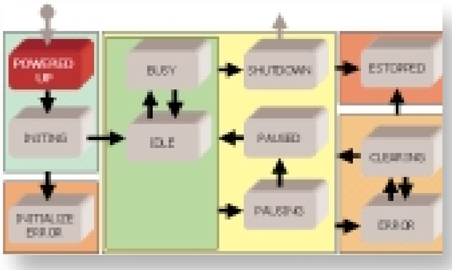

Fig. 6: Standard state diagram (based on ASTM E1989-98 Standard)

nance periods, to prevent system crashes. If this happens, however, activities have to be able to side-step to similar devices, or the samples affected must be ‘parked’ until the maintenance work is complete.

The scheduler is linked to a path-planning module which has still to be integrated. This recognises potential collisions with stationary and moving obstacles in advance and calculates an alternative progression pathway for the robot used.

Together with the dispatcher, which monitors all activities automatically, and the path-planner, the scheduler is one of the most important components in the WICIL Controlling Architecture (WCA) [4].

Accessing devices, raw data and processing of results

The executor accesses the devices, receiving input from the scheduler in the form of the so-called execution plan. The latter is a time-based summary of the non-interruptible sequences of activities which have to be performed on each device and the individual activities. One of the important functions of the executor is to plan the exact timing of activities and the device-specific monitoring of the allotted times. If an allotted time is not adhered to, the executor has to transmit this information to the system, so that each component affected can react in the appropriate manner.

The WCA is basically designed so that the system must be stopped only in extreme emergencies and so that foreseeable exceptional situations are solved automatically. The objective is to ensure as far as possible that no samples are made unusable or raw data or results are lost (see the GENIUS module for more information on this).

As already mentioned, the devices are accessed via CORBA. Amongst other things, this means that devices linked with computers other than the executor computer can be incorporated into the operations on the laboratory bench.

The devices are accessed using a standard mechanism, which is largely based on the ASTM standard Lecis (Laboratory Equipment Communication Interface Standard, ASTM E1989-98) (Fig. 6) [5] [11] [12] [14]. At all times, each device is in a well-defined state, which permits only certain subsequent transitions to other states (arrows in Fig. 6).

After powering up and initialising, the device runs in the idle status (IDLE), and the next command in the execution plan can be executed from this state (BUSY). The commands available for a given device are extracted from the SCD when the protocol is prepared and are inserted into the workflow. They are then executed in this phase.

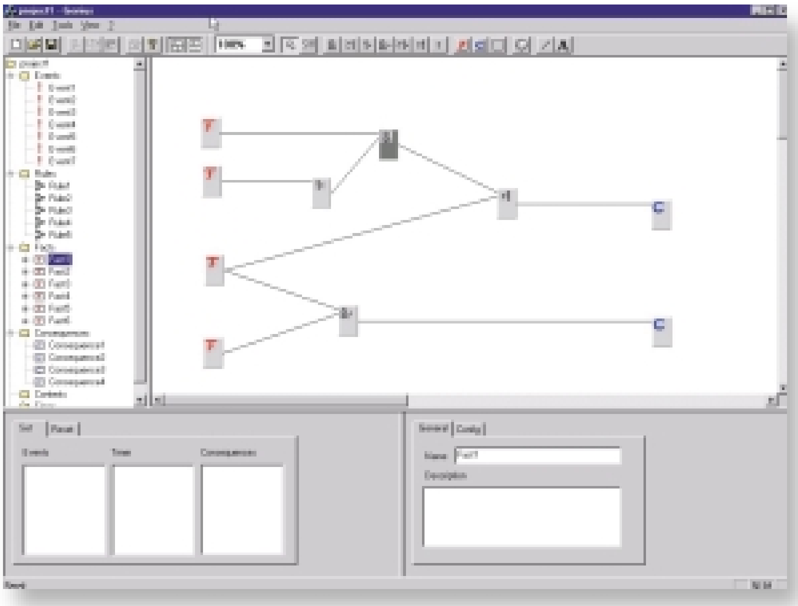

Fig. 7: Graphic definition of rules in GENIUS

In addition to “normal” commands, there are commands for exception handling. For example, the workflow can be interrupted or a shutdown procedure activated from all states shaded green. An emergency stop (Estop) can be activated from all states (shaded yellow and Error), but from this state the system can only be powered up again manually.

Automated processing of events and exceptional states (GENIUS)

Exceptional states are at present hard-coded in the programme code. Devices can “input” their own exception handling if they have control and evaluation programmes on board. Exceptional states caused by the interaction between components on the laboratory bench cannot be hard-coded, because the devices may be from different manufacturers. Processing of exceptional states by the WICIL Controlling Architecture is performed via the expert system module GENIUS (Generic Events Network Improves Unattended Systems) [17].

Exceptional states are represented by events, which are available to all registered system components. Combinations of events form rules, which "trigger”, provided that all necessary facts for this to happen are present. The WICIL GENIUS package can be used to process any system functionalities which produce events [20] [22].

The rules are represented graphically (Fig. 7). Several events can be grouped to form a fact, can have time conditions attached, and can be linked via Boolean operators. The facts can be reset by events or consequences, and also by consequences of other rules. The consequences are also events, with which functions or dialogs can be launched, amongst other things. In this way, suitable measures can be triggered if, for example, an exception occurs.

Summary and outlook

In principle, it has now been demonstrated that interruption-free automated operation on the laboratory bench is possible. All models described are fully operational, but have not yet been fully integrated. One of the reasons for this is that the CORBA specification has not yet been completely implemented. Of particular importance is the functionality of the system capability dataset, with which it has become possible for the first time to standardise laboratory device interfaces, without restricting the wide range of functions of different devices. This is guaranteed by the meta-description approach, i.e. the laying down of the notation description and not the notation itself. In doing so, the aim is to achieve complete standardisation. At the Wiesbaden Computer-integrated Laboratory, we would welcome more extensive support from device manufacturers and users. Only in this way will we be able to fully implement all of our concepts.

Finally, we should like to take this opportunity to thank the State of Hessia and in particular Hewlett Packard Deutschland GmbH and the Creon-Labcontrol AG for their material and intellectual support. It was they who made it possible for us to put our ideas into practice.

Literature

[1] R. Schäfer, LIMS als Komponente der Informationsarchitektur eines Unternehmens, CHEManager 4 (1996) 24-26

[2] R. Schäfer, B. Stiehl, Expert System for Dynamic Scheduling Supporting Sample Preparation Constraints, Poster, Int'l. LabAutomation Conference, San Diego, California, USA, 1997

[3] R. Schäfer, M. Hagemeier, M. Hofmann, K. Reubert, A Highly Distributed Laboratory Automation System Using the CORBA Standard as Integration Platform, Int'l LabAutomation98 Conference, San Diego, CA, USA, 1998

[4] R. Schäfer, T. J. Beugelsdijk, J. Elling, Architectural Issues for Total Lab Automation, Int'l LabAutomation98 Conference, San Diego, CA, USA, 1998

[5] B. Schäfer, T. Staab, R. Schäfer, J. Elling, Standardized control of laboratory equipment with the American Society for Testing and Materials (ASTM) Laboratory Equipment Control Interface Specification (LECIS), Int'l LabAutomation98 Conference, San Diego, CA, USA, 1998

[6] C. Piotrowski, T. Richter, R. Schäfer and G. W. Kramer, System Capability Dataset for Laboratory Automation System Integration, Journal of Analytical Laboratory Automation, 12 (1998)

[7] O. Borchert, G. W. Kramer and R. Schäfer, Dealing with Result Data Automatically - What Does a Capability Dataset Need to Contain?, Int'l LabAutomation99 Conference, San Diego, CA, USA, 1999

[8] T. Tauber, G. W. Kramer and R. Schäfer, Device Integration Using the Device Capability Dataset Concept, Int'l LabAutomation99 Conference, San Diego, CA, USA, 1999

[9] R. Schäfer, Advanced Scheduling and its Impact on Laboratory Integration, Int'l LabAutomation99 Conference, San Diego, CA, USA, 1999

[10] – [17] Author will send references by request

Professor Reinhold Schäfer
Degree in physics and doctorate at the University of Mainz, worked in the pharmaceutical industry, last position as group head for scientific information systems in a company with 24,000 employees, realisation of various automation projects. Since 1989, Professor for Databases, Artificial Intelligence and Laboratory Automation at

Fachhochschule Wiesbaden – University
of Applied Sciences
Computer Science Department
Kurt-Schumacher-Ring 18
D-65197 Wiesbaden, Germany
Fax 0611/9495-210
E-mail: schaefer@informatik.fh-wiesbaden.de