FUJITSU

WHITE PAPER

Job scheduling with AVAS in BS2000

Issue August 2016

Pages 16

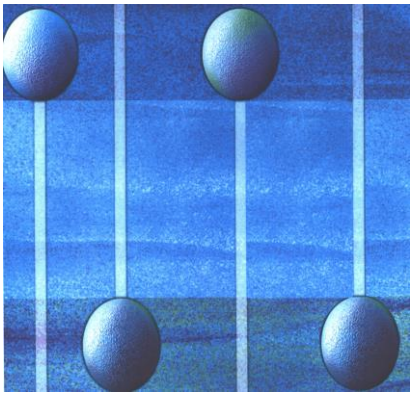
The complexity and workload of data centers are constantly increasing.

This gives rise to the demand in IT operations for constant productivity gains and improved service quality as a result of clear structuring and a high degree of transparency and flexibility.

Cost-effective administration, efficient use of operating resources and 24-hour operation are possible only if all essential operating processes can be automated. One of the key factors is the automation of the batch production.

The AVAS job management and scheduling system is a product that enables users to automate their job production.

A single AVAS system on BS2000 can schedule and monitor jobs of several BS2000 systems and of heterogeneous open systems servers.



Contents

Overview

The complexity and workload of data centers are constantly increasing.
This gives rise to the demand in IT operations for constant productivity gains and improved service quality as a result of clear structuring and a high degree of transparency and flexibility.
One of the key factors in achieving this goal is the automation of all recurrent or predictable operations, in other words the automation of batch production.
The AVAS job management and scheduling system is a product that enables users to automate their job production.
A single AVAS system on BS2000 can control and monitor jobs on several BS2000 systems. The workload can be distributed dynamically between BS2000 systems.
With AVAS, users can automate job production with the optimum utilization of capacities to such an extent that dialog input is reduced to a minimum. By this AVAS smoothes the way to transferring job production to unmanned shifts.
Information about error situations is delivered to the BS2000 console interface. By this monitoring products may cause an alarm or call the stand-by service.
AVAS systems can be monitored from a single point via their SNMP interface using an SNMP-compatible network management facility.

The AVAS job nets constitute an easy-to-use means of defining

- when
- which jobs are performed
- with what modifications
- in what order
- on which system
- with which dependencies
- and using what procedures in the event of an error.

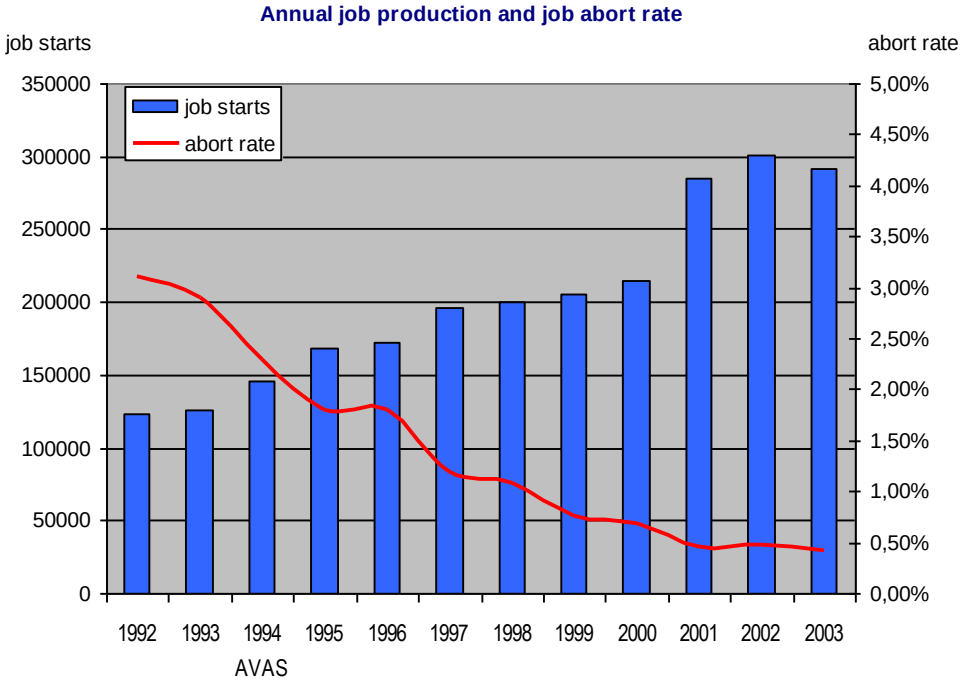
Existing jobs and procedures can continue to run unchanged. It is no longer necessary to create complex control procedures that are difficult to maintain.
Input data for job preparation (job parameters) can be supplied automatically or in mask-driven dialogs.
Deadlines and planning periods can be specified in symbolic form.
One or more calendars are provided, mapping symbolic representations on real dates, e.g.
- ultimo on the last working day of the current month.
The standard AVAS calendar contains all the usual important dates.
This description can be used to fully automate recurring processes by AVAS.
Ad-hoc requirements are planned in a dialog, where AVAS offers easy-to-use options for transferring the creation of jobs to the relevant non-DP departments, if required.
By combining AVAS with the product MAREN (volume management system), it is possible to ensure in advance that all volumes required by planned jobs are available, and that the actual volume identifiers are automatically entered in jobs. In addition, volume readying lists can be produced for the archive and the operator.
AVAS performs event driven. By this the job nets are being processed without delays and there is no consumption of resources by cyclic request processing.
AVAS does not require any special privileges (TSOS).
Modifications to planned processes and repairs to faulty jobs can be carried out directly in the AVAS system with the appropriate authorization.
During this process, the database of the job production remains unchanged, because on principle AVAS uses job copies when processing the job production.
An access control facility protects the system against modifications by unauthorized users, and allows for the organizational separation of task areas.
All input data, modifications, and results are logged in the AVAS journal, allowing users to check back over all operations performed.
The graphical PC interface can be customized as desired.
The relations between AVAS objects and job objects can be stored and queried in an SQL database (e.g. on the PC).
In addition to a modern command language, PC-supported CASE tools are available for the development and maintenance of jobs and procedures.

AVAS benefits

Over the past years, the workload involved in operating data centers has increased significantly in volume and complexity. To reduce the resulting quality risks and the drastic rise in personnel costs, computerized procedures are now being used to automate operating and scheduling procedures.

The introduction of AVAS offers the following advantages:

- Productivity increases in job management and scheduling:
  Through interactive, bulk-oriented operation the workload that can be managed and handled by each scheduler and operator is increased significantly.
- Deadlines met consistently:
  Jobs are scheduled in accordance with the deadline specifications laid down by customers (e.g. non-DP departments). It is thus possible, with no increase in complexity, to make maximum use of the existing parallelism of processes and thus of the time limitations of the batch window.
- Performance monitoring and trend analysis:
  The important quantities of the batch production as medium, maximal and minimal run times are stored and may be queried or evaluated at any time.
- Enhanced quality and reliability in job scheduling:
  Error sources (e.g. complex dependencies between individual jobs or between jobs and other events) and error-generated problems in connection with jobs or hardware/software components are reduced.
  A closed production environment prevents illegal or inadvertent intervention in production.
  In practice, it has been demonstrated that the production error rate due to failed jobs can be significantly reduced (e.g. from 5% to less than 1%).
- Great transparency:
  Thanks to the fully integrated logging and documentation facilities, the execution, modifications and results of all jobs can be easily reconstructed at any time for customers, schedulers, operators and controllers.
- Further productivity increases in job processing:
  When using the AVAS batch functions the job planning, preparation and release is fully automatic.
- Reduction in manned shifts:
  The options for starting production automatically at predefined times and for restarting production following failures allow data centers to cut back on the number of manned shifts.

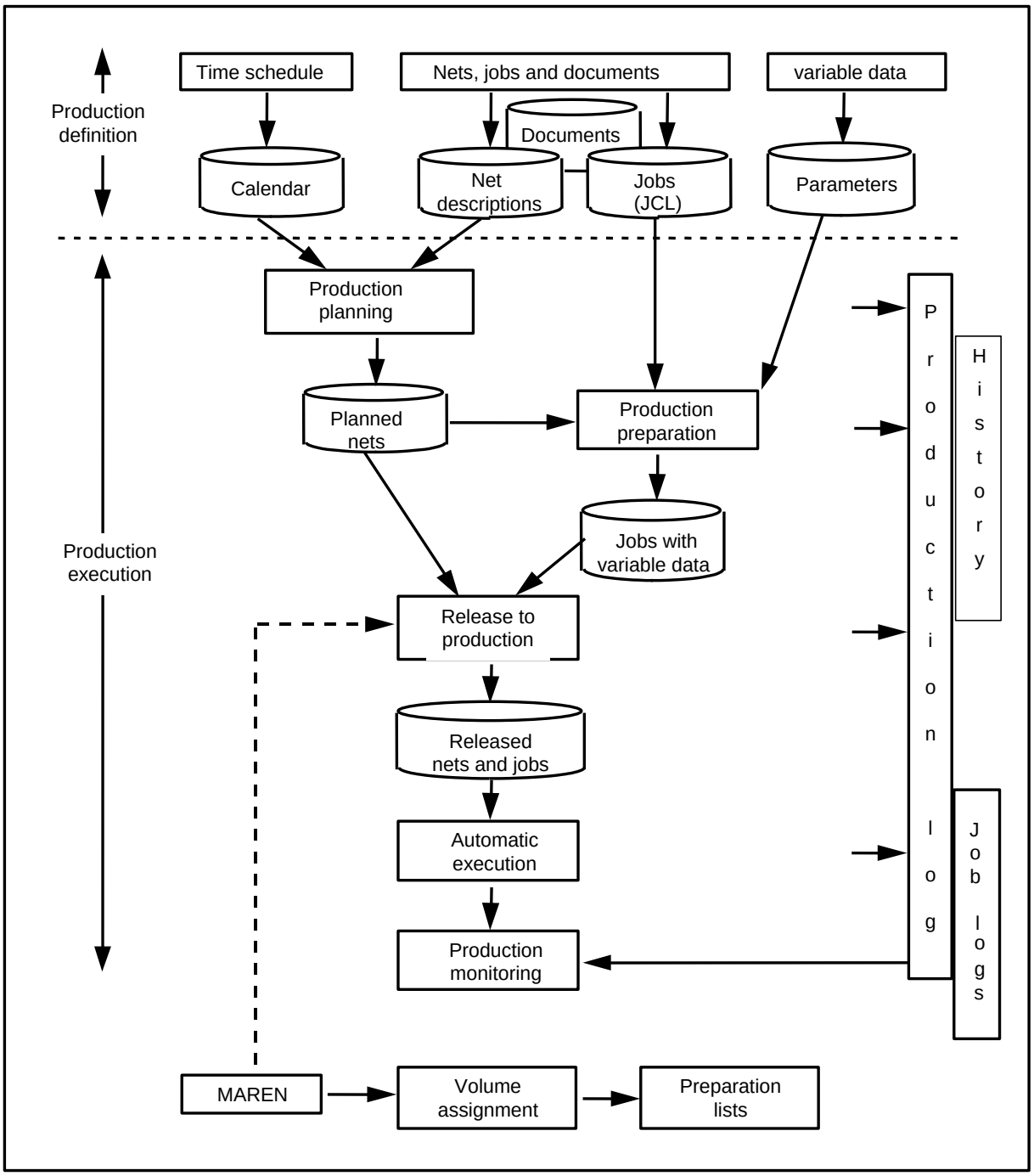


The diagram above shows a typical AVAS installation.

- Quality (the number of unplanned job aborts) has been improved by a factor of more than 6, dropping from 3.1% to below 0.5 % today.
- The job volume has increased by more than 100%.
  With AVAS, these improvements have been achieved without changing the size of the workforce.

By using AVAS for automatic processing, also the increasing number of procedures in the heterogeneous IT environment takes benefit of the quality improvement and productivity gains, which can be achieved by AVAS.

AVAS functions overview



AVAS — Schematic function diagram

The AVAS functions are subdivided into the following two areas:

- Production definition
  i.e. the provision of data such as net and job descriptions, parameters, and the time schedule for processing under AVAS
- Production processing
  i.e. implementation of the production runs at the scheduled times on the basis of the data provided, the logging of all actions, and the transfer of job logs to AVAS management.

Production definition

Logically related jobs are combined together to form job nets for execution under AVAS.

A job net consists of 1 to n BS2000 jobs. BS2000 jobs can be both conventional ENTER jobs and SDF-P S-procedures.
Jobs and job nets in turn can be combined as hypernets.

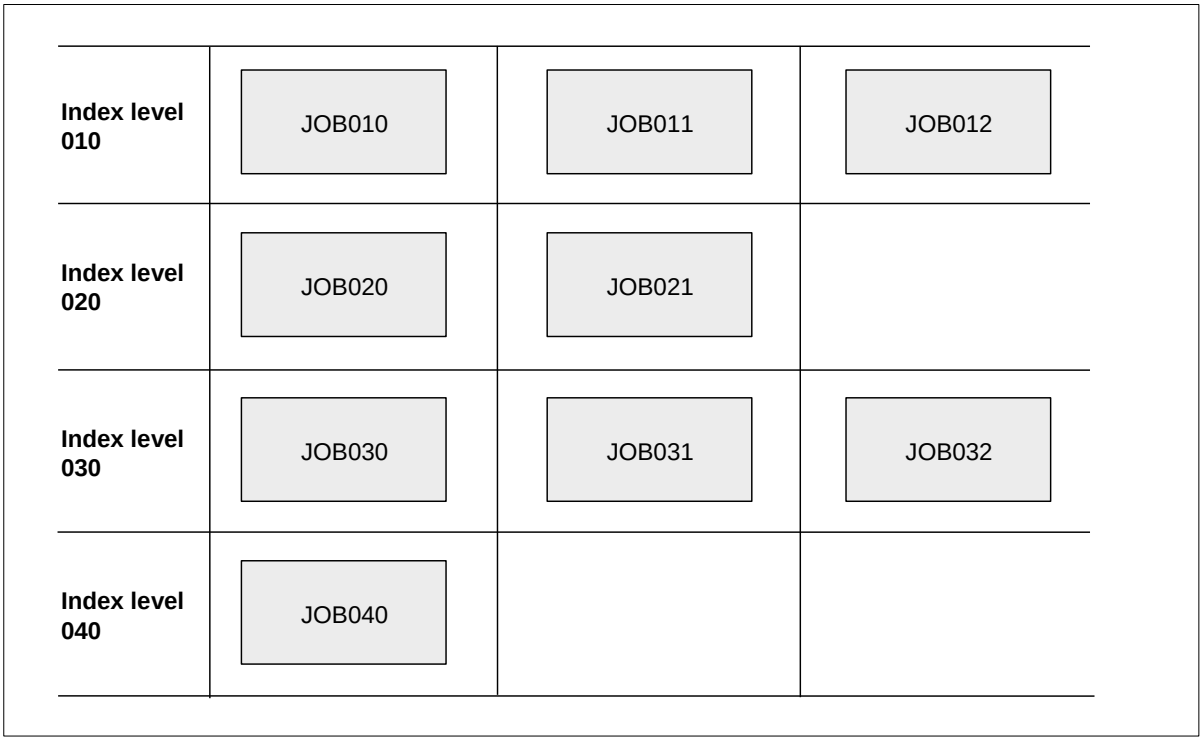
Instead of complex controls that define the sequence of jobs, AVAS forms transparent nets that take account of all logical and temporal dependencies across all platforms.

Within each job, it is possible to insert parameter placeholders for variable job data.
Using a special statement in the JCL, the user can request AVAS to transfer job-specific execution data (such as the job log) to AVAS management.

The AVAS job net

The processing sequence and the requirements for starting individual steps are defined in the net structure. Under AVAS, the arrangement of jobs within a net is controlled by index levels.

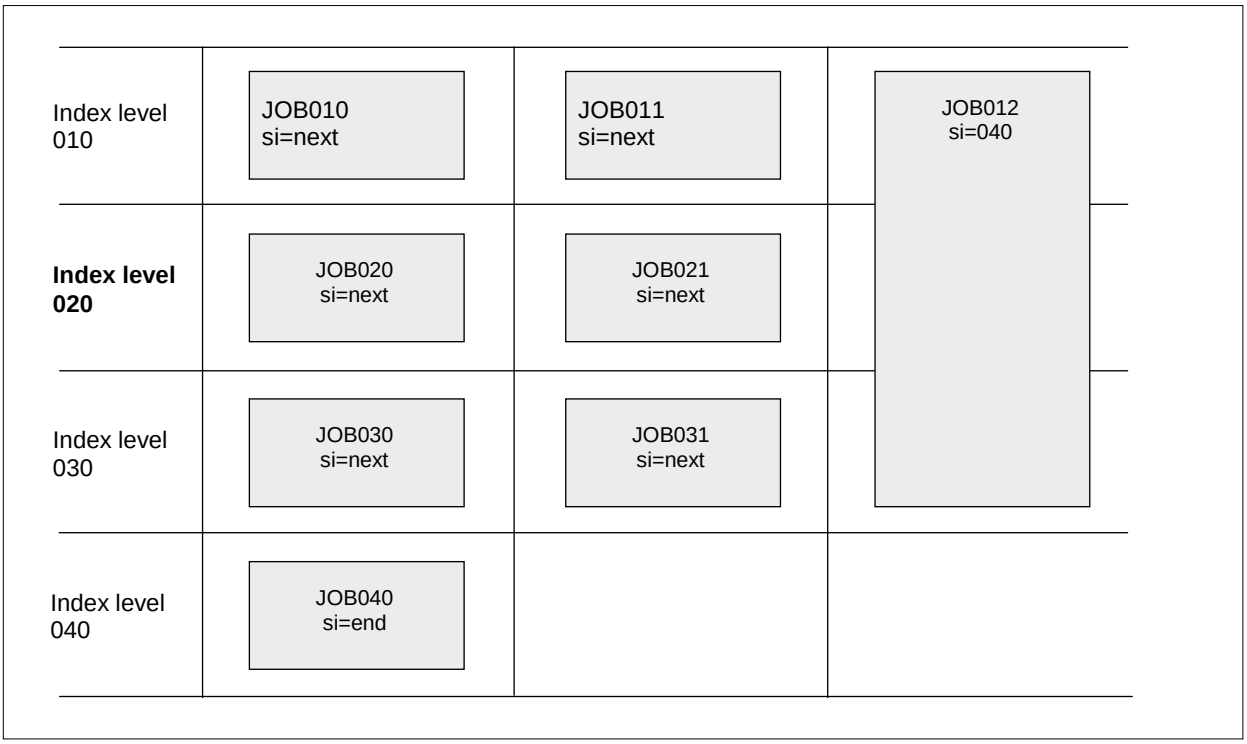
All jobs on the same index level are started in parallel. The individual index levels are executed sequentially, i.e. the jobs on a given index level are not started until all the jobs in the preceding level have terminated without errors



AVAS — Sample AVAS job net

In this example, JOB020 and JOB021 are not started until JOB010, JOB011 and JOB012 have terminated without error.

To exploit the existing parallelism, it is possible to use the synchronization index of a job in order to specify the index level at which the net is to wait for the end of a job.



AVAS — Sample AVAS job net with synchronization index

In this example, the specification of synchronization index ‘si=040’ means that JOB012 can run in parallel to all jobs apart from those on the final level 040.
All other jobs are started sequentially in accordance with the default value ‘si=next’ as shown above.

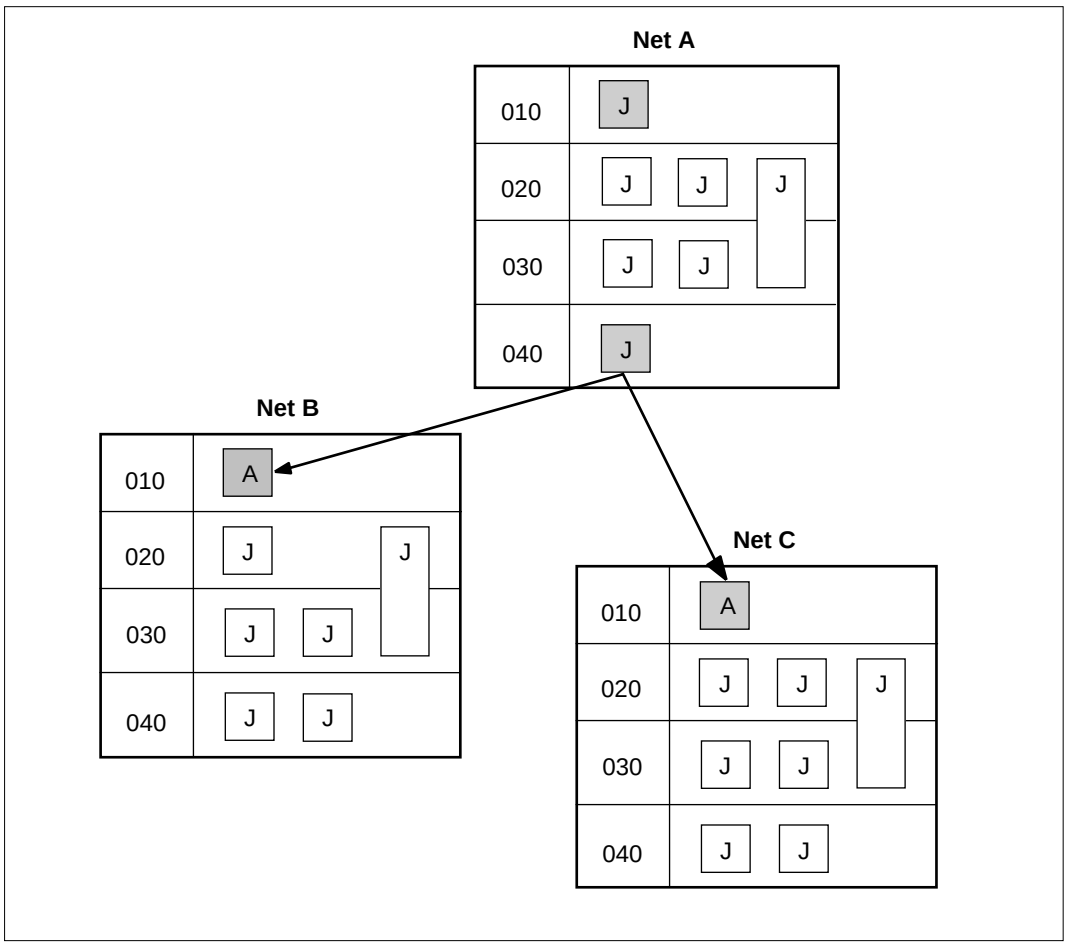
The AVAS CHECK function helps the user to define consistent parallel paths in the AVAS nets.

Dependencies

Within the net structure, dependencies are specified in the same way as jobs.

These may be dependencies on other nets, jobs, conditional values, or resources.
The following condition types can be described and queried in the net description:

Condition type JOB	Dependency on a job in the same net or in another net
Condition type JVA	Dependency on the predefined value of a job variable
Condition type NET	Dependency on another net
Condition type RESOURCE	Dependency on the availability of a resource
Condition type TIME	Dependency on arrival at a predefined time
Condition type VALUE	Dependency on a conditional value



AVAS — Example of inter net dependencies

In the example above, net B and net C wait for the normal termination of net A.

This result in the following sequence: net A and subsequently nets B and C

The concept for defining and querying dependencies on freely selectable resources (condition type RESOURCE) and conditional values (condition type VALUE) opens up the main dimension in mapping organizational factors.
On a logical level, AVAS nets can be interconnected at any point in a process, and dependencies on objects and events outside AVAS can be created.
The condition types RESOURCE and VALUE can be defined, modified, deleted or queried in the net structure. They can thus be made available for use both by the net itself and by other nets.
Conditional values can be linked by means of logical operands.
Resources are allocated and released on an exclusive or shared basis.

Documentation elements can be defined for the documentation of nets, jobs and dependencies.

The net restart feature is provided for recovering from error situations in nets. The desired runtime variants for the net restart can be predefined in the net structure.

Hypernets

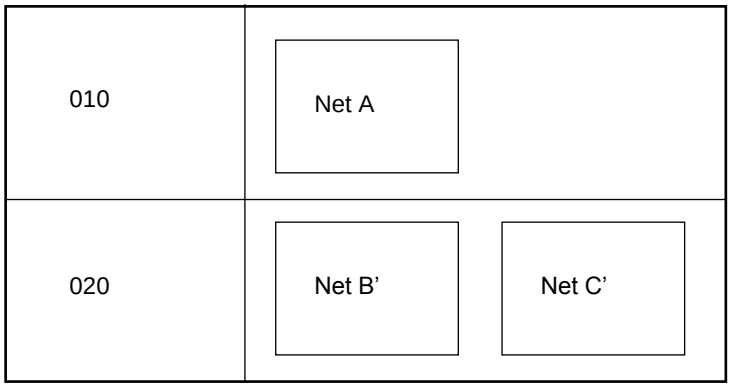
Groups of nets and jobs can be combined into a so-called hypernet.

The planning, production preparation, and the release of the subnets, which means the job nets of the hypernet take place via the hypernet.

When usings hypernets the dependencies between jobs and job nets can be defined transparently in the superior hypernets and need not be defined inside the job nets by net-end conditions.

With hypernets, the production sequence can be defined notably simple and transparent.

Production control is arranged clearly, because only the larger unit hypernet has to be observed instead of single job nets.



AVAS Sample Hypernet

The hypernet combines the job nets of the above example “inter net dependencies “ .
The job nets B’ and C’ differ from B and C by the fact that the net dependencies (on index level 010 of the nets B and C) are omitted – as they are no more necessary.

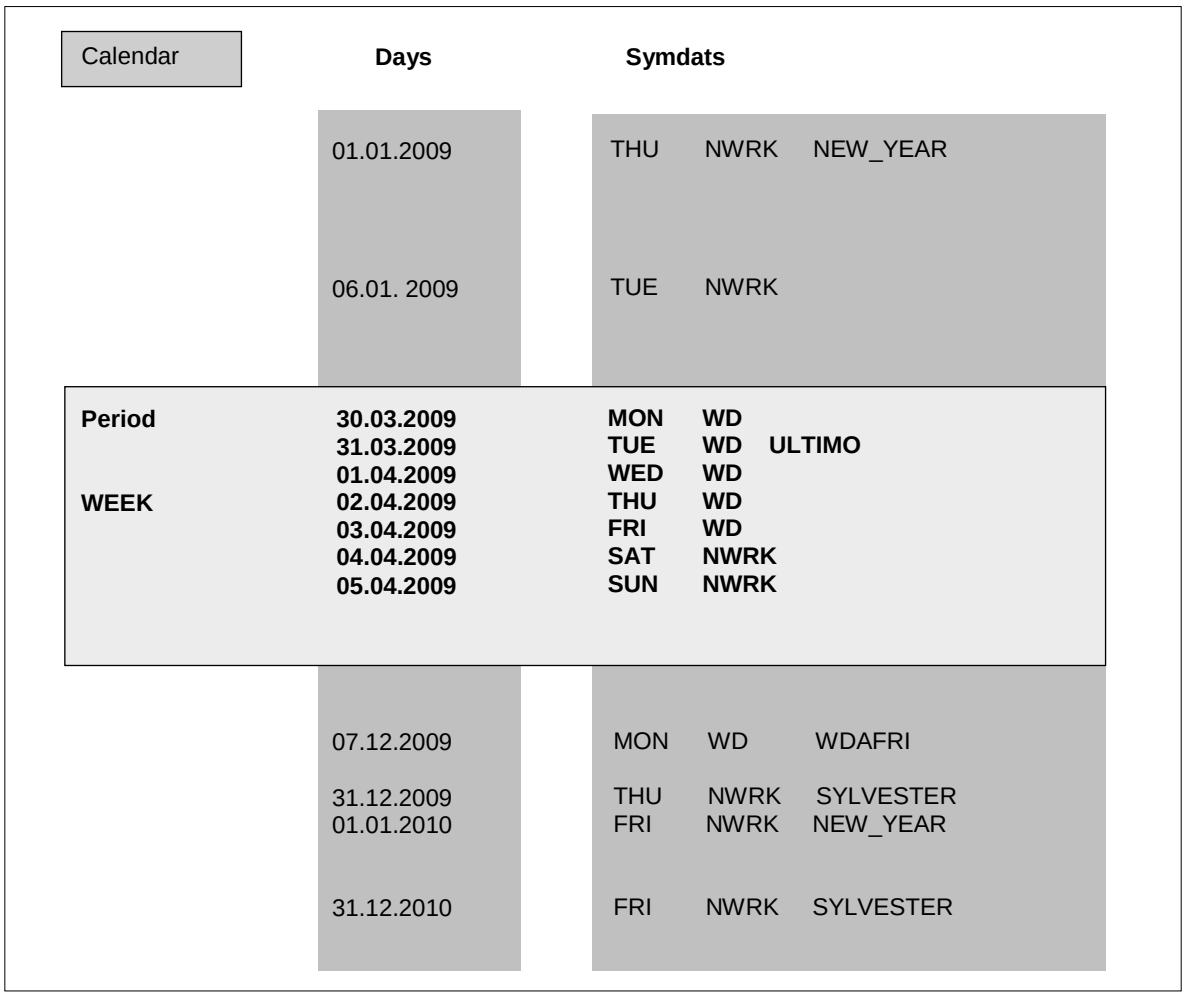
The AVAS calendar

Scheduling

All scheduling activities are performed using calendars that extend over several years and include a day column containing real dates and symbolic dates – the so-called symdats assigned to individual calendar days. To distinguish from working days, days can be declared as non-working days (NWRK) or free days. This forms the basis for typical exception rules such as WDPFRI ='working day following (past) a Friday that is not a working day'.

AVAS is supplied with a calendar that already contains all the usual important dates in symbolic form.

The calendar and the job nets are linked by means of symbolic dates. In the net description, symbolic start dates can be assigned to the net as a whole or to individual jobs or dependencies in the net. During production planning, the calendar days are evaluated and the symbolic dates entered are used to determine which nets are to be executed by this date.



AVAS — Sample calendar containing symbolic dates (symdats) and a period

To facilitate the long-term planning of production runs, it is possible to define periods that specify a planning area within the calendar.

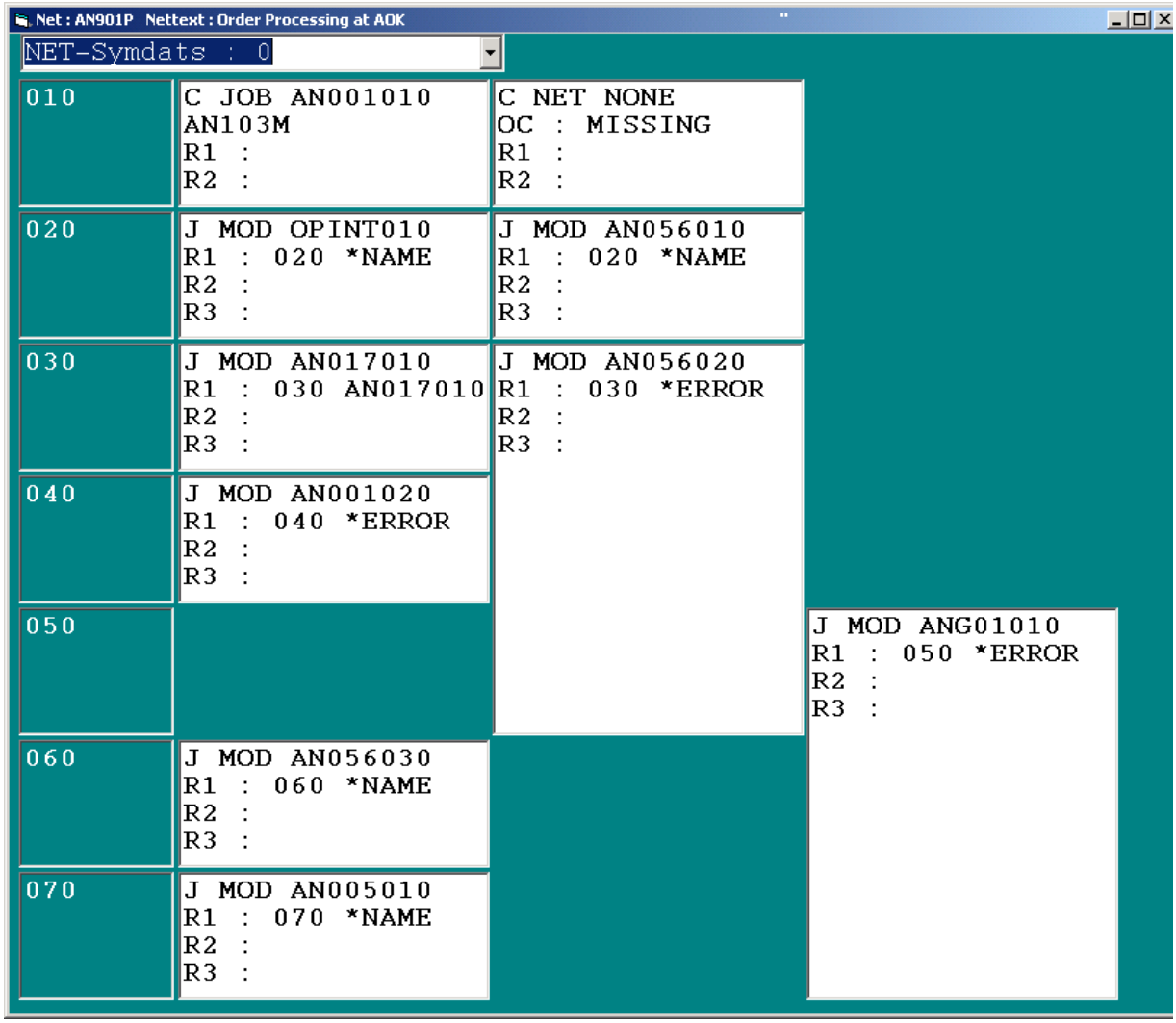
With AVAS, all the normal periods, such as TODAY, (this) WEEK, etc. are already predefined.
They are updated automatically by AVAS on a daily basis, and their identifiers are freely selectable.

This provides the foundation for the flexible planning of IT production

AVAS-QUER utility routine

The AVAS utility routine AVAS-QUER reads the AVAS master database and selects data for further processing in relational databases, e.g. on a PC.
Using the generated database tables, cross-references between various AVAS objects can thus be easily queried, for instance:

- Which net is waiting for the job variable/resource/net/job xy?
- Where is job xy used?
- Which nets are running on symdat K01?
- What is the structure of net xy?
- Which jobs/nets are started on system xy?



AVAS - example using AVAS-QUER-data, an application displays job nets graphically

Production processing

Production processing is made up of the following steps:

- Production planning
- Production preparation
- Release for production
- Automatic production execution
- Production monitoring

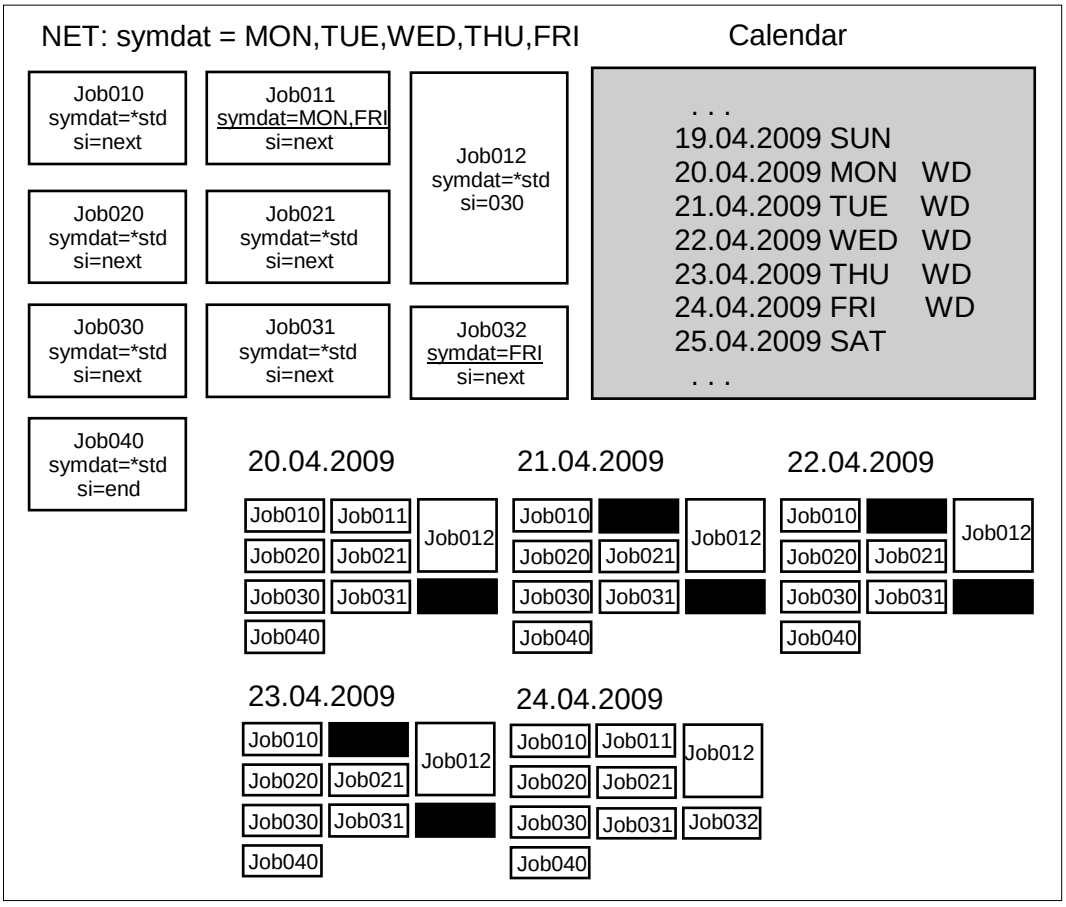
Production planning

The freely selectable period for a forthcoming production run can be defined either using the calendar, or in the case of individual nets, without the calendar.

During production planning, AVAS nets are linked to real start dates, supplied with variable data for the actual production run, and incorporated in the production plan.

Using symbolic dates, it is possible to form subnets (known as net variants) from a net or hypernet. A net element (subnet, job, condition) is only planned into the production run if the symbolic start date of the net is contained in the list of symbolic dates of the element.

Thus, a **single** net can be used for different runs on different days, as illustrated by the following example:
on Mondays and Fridays, Job011 is to be run in addition, and on Fridays Job032 is to be run.



AVAS — Example of production planning with net variants

Further definition possibilities arise from the feature of linking symbolic start dates using an AND or NOT operation.
Linking of symbolic start dates using AND and NOT is possible with the symbolic start dates for

- selecting the nets to be planned and
- selecting the net elements (subnet, job, condition) of the nets to be planned.

From the AVAS production plan, it is possible to determine which nets are run on which days, and their current stage of preparation.

Production preparation

The main purpose of production preparation is to provide the current input data for planned jobs, i.e. to update values that vary from job to job (parameters).
These values are either AVAS or procedure parameters entered in the JCL. Escape characters define the positions at which parameters should be entered in the jobs.

AVAS offers the following options for assigning current values to net or job parameters:

- mask-driven assignment in a dialog
- values in parameter files
- user and system variables managed by AVAS, such as ‘Company’, Date and Time

The options of using a parameter file or variables managed by AVAS support fully automatic production scheduling, e.g. where the user automatically supplies error-free, up-to-date values through upstream computerized procedures.
In the case of manual assignment, the mask-driven dialog minimizes the risk of typing errors.

AVAS keeps preparation separate from production, i.e. preparation can complete the tasks assigned to it via AVAS asynchronously to production.
In this way, a net that is to run more than once can be supplied with the appropriate parameters in advance for each separate date.

Release for production

Net processing starts as soon as the nets are released either automatically or explicitly by production preparation.
It is also possible to release single nets for processing.

Combination of AVAS and MAREN

If the interfacing module to the MAREN volume management system is used, it is possible to automatically ensure during release that all volumes required by planned jobs are available, and that the current volume identifiers can be entered in jobs. In addition, volume readying lists can be generated for the archive and the operator.

Automatic production execution

Running nets

The production execution is initiated and controlled by the scheduling component of AVAS, the run control system.

After release, the individual jobs are scheduled automatically.
The run control system starts the nets at the specified times, queuing the jobs in the order defined in the structure description and taking into account all declared dependencies on other nets, jobs, job variables, conditional values, and resources. In the process, AVAS monitors the normal and abnormal termination of each individual job.

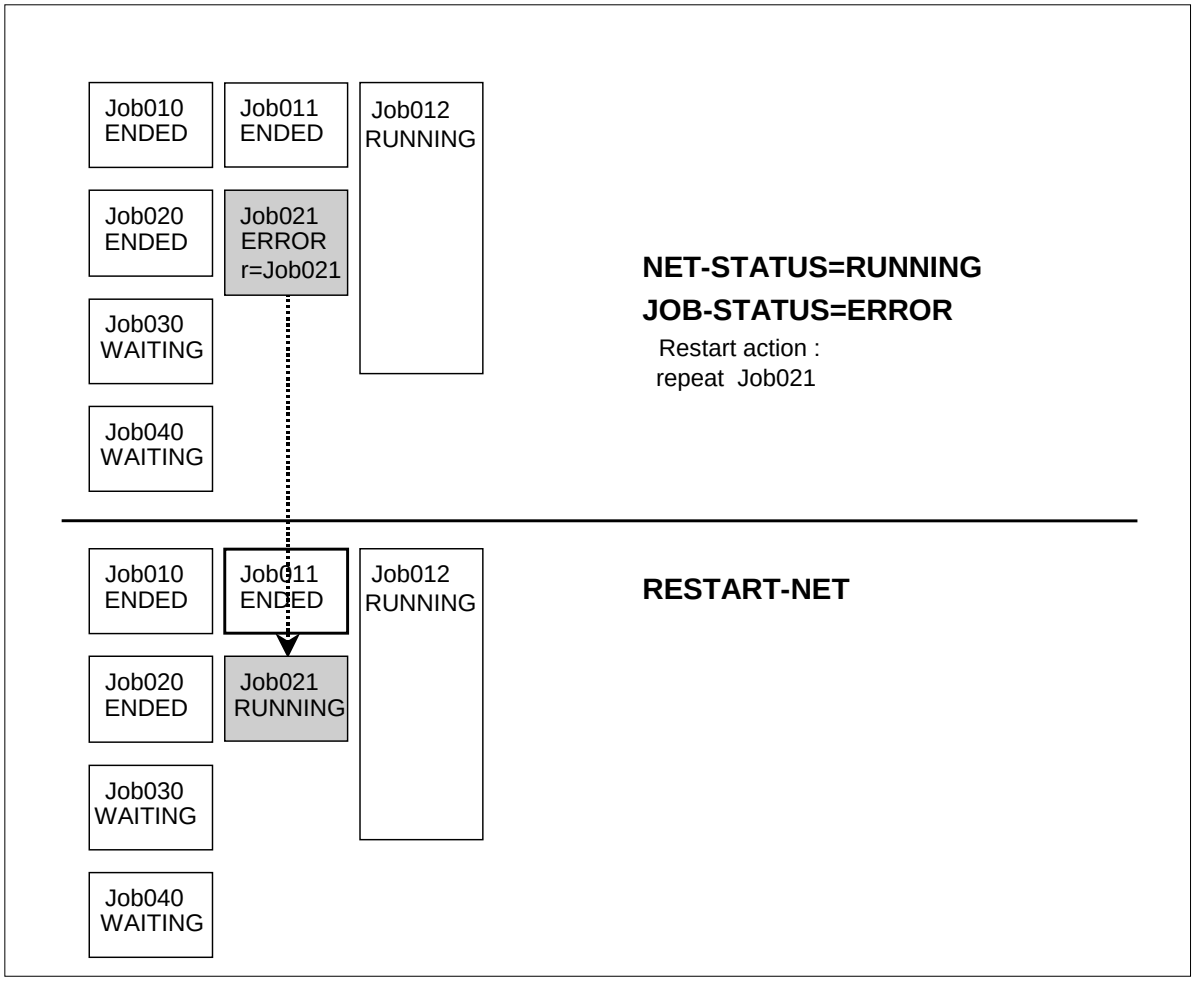
To cater for situations where nets or jobs do not start or dependencies cannot be resolved in good time, it is possible to define a maximum waiting period. Once this period is exceeded, AVAS initiates the actions defined by the user (e.g. abort).

Restarting nets following error situations

If a net aborts due to an error situation in one of its elements (jobs, conditions), it is possible to restart the net.

The restart variants defined for this purpose in the net description determine the point at which processing resumes.
In the event of a restart therefore, individual jobs may be skipped or additional jobs may be executed. The parallel structure defined in the net is not affected by the restart, i.e. even if the net is still running, a restart can be activated for independent net branches.

A restart can be initiated either automatically by AVAS, or at the request of the user after checking the error situation.



AVAS — Example of a restart in an active net

In the above example, JOB021 encounters an error. This job can be restarted immediately, i.e. while other jobs (JOB012) are running in a parallel net branch.

Abnormal terminations do not necessary result in a negative status. The user can define which status leads to compliance with or violation of a condition. If there are no user-defined values, AVAS uses default values. Compliance with or violation of a condition can thus be programmed in advance

Production monitoring

Net processing can be monitored at all stages of the production run. Thus, the progress of an individual net or of the entire production run can be observed. Similarly, monitoring can also encompass all nets with a certain processing status. The documentation associated with nets and jobs can also be displayed.

Monitor

Driven by events, the start and end of a net are displayed, together with any faults that occur in the process, i.e. when net elements are unexpectedly placed in a waiting state due to error situations or non-compliance with conditions.

Journal

For control purposes the monitoring of all DP production runs handled by the AVAS system is based on the production log, also known as the journal file.
It contains all user activities and all actions of the AVAS system for the processed nets.

Runtime logs

AVAS can take over the runtime logs of all jobs that run under its control, and manage these together with the jobs themselves. In the event of restarts, there may be several runtime logs for one particular job.
In addition to the actual "SYSOUT" runtime log, it is also possible to transfer other log data to AVAS.
The job whose runtime log is to be transferred assigns the log to a cataloged file. It then calls a signaling program which issues

a request to AVAS to retrieve the appropriate runtime log.
The job logs taken over by AVAS can be displayed directly from the AVAS dialog.

Reports

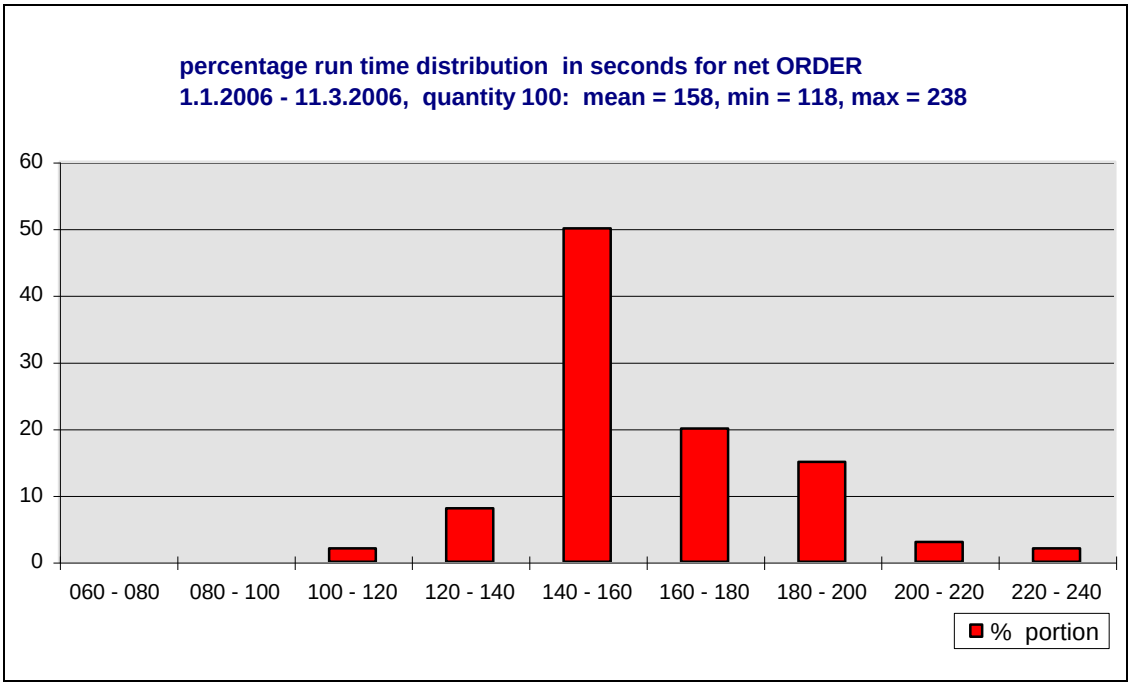
Reports are used for the retrospective monitoring of production.
For example, in the case of daily inspection, the report function generates the following standard reports:

- PLANNED-NET-MODIFICATION report containing all unplanned modifications to nets and jobs, which were carried out after planning
- OUT-OF-PLAN report containing all planning deviations that occurred during net handling, e.g. delays and error situations

Performance monitoring and Trend analysis

The important quantities of the batch production as medium, maximal and minimal run times may be queried online or displayed at the PC. This facility is based on a file, into which AVAS stores compressed historical data – the History-file.

The facility provides the comparative evaluation of recurrent runs (performance monitoring and trend analysis) and the forecasting of run behavior in the future.



AVAS - Example - Presentation of History-data

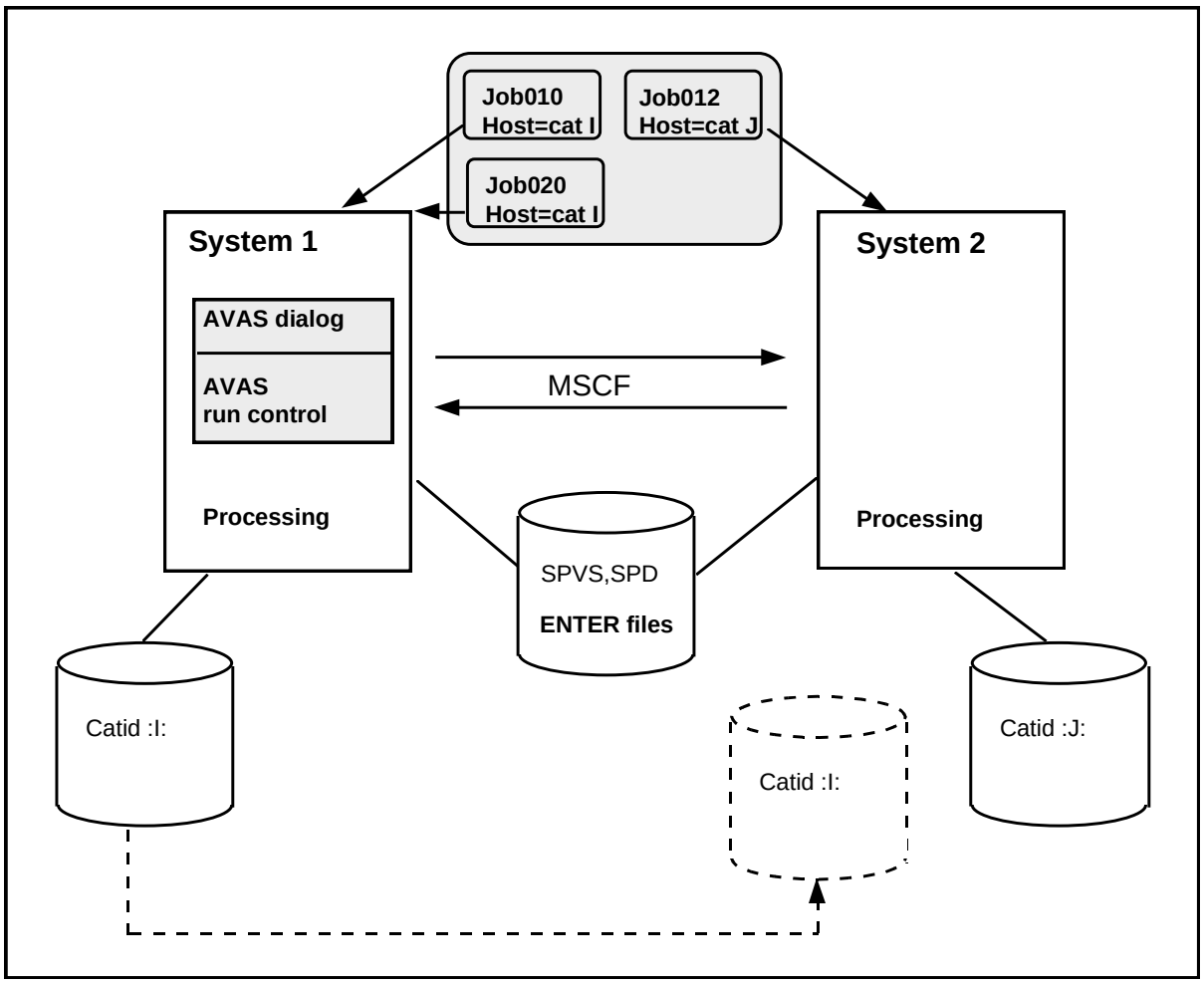
BS2000 multi-host operation

With the HIPLEX MSCF multi-host function, it is possible to connect several BS2000 Business Servers to a fail-proof computer network (HIPLEX cluster).

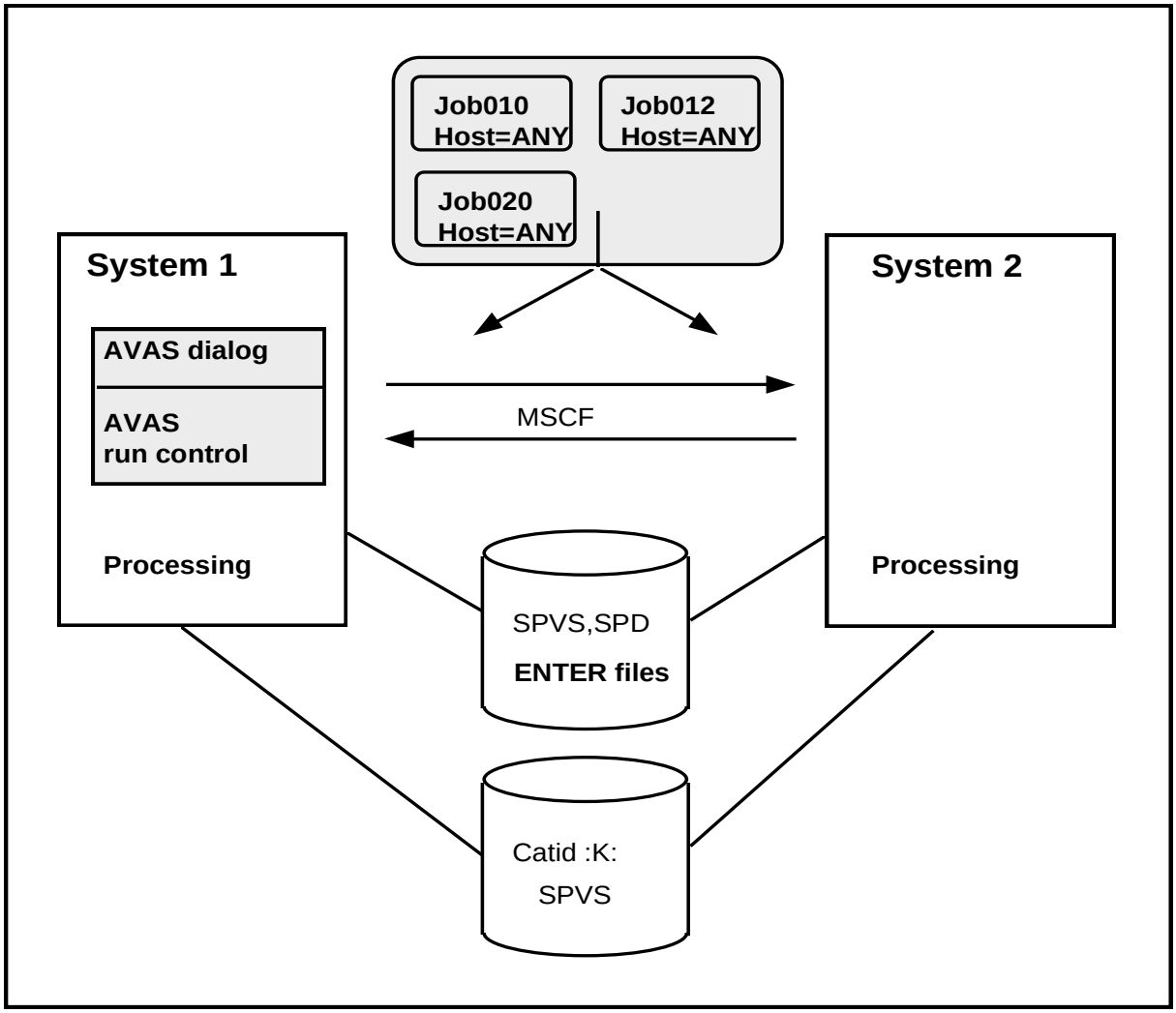
In AVAS multi-host operation, one particular host is assigned the role of AVAS master. This host runs the AVAS system, including the run control system. AVAS can distribute entire nets or individual jobs within nets to any host in the MSCF network, and then monitor the processing of these distributed jobs. A shared disk ensures that the job (ENTER) files can be accessed from any host.

Thanks to the distributed handling of nets and jobs in the MSCF network, it is possible to ensure optimum load distribution, while automatically taking into consideration any logical or temporal dependencies.

Following a system failure, the workload involved and (in the case of the AVAS master system) AVAS itself can be automatically relocated to an active host.



AVAS - BS2000 multi-host operation



AVAS - BS2000 multi-host operation – load-dependent dynamic job distribution

In the case of jobs that can run on any host in the network, i.e. whose data (here catalog ID K) can be accessed from any host, the specification of the destination system may be left open (HOST=ANY).

AVAS ensures that these jobs run on the system with the lightest load.

In the case of a long distance between the BS2000 servers, the remote BS2000 can be linked via an AVAS-SV-BS2 server alternatively, which starts the BS2000 job, monitors the sequence and reports the result back to the AVAS Master. The job can be processed on the local and on the remote BS2000 without modification.

Published by department:

Barbara Stadler
Phone: ++49 (0)89 620601978
Barbara.Stadler@ts.fujitsu.com
ts.fujitsu.com