A Domain Description Language for Job-Shop Scheduling

Howard Beck, Ken Currie and Austin Tate

AIAI-TR-137

October 1993

This work has been undertaken as part of the TOSCA project with the support of Hitachi Limited.

Artificial Intelligence Applications Institute
University of Edinburgh
80 South Bridge
Edinburgh EH1 1HN
United Kingdom

© The University of Edinburgh, 1993.

Abstract

As part of the TOSCA project undertaken for Hitachi Limited, the Artificial Intelligence Applications Institute at the University of Edinburgh has produced a Domain Description Language for job-shop scheduling. The DDL enables the testbed dataset (based on real factory data) used in that project to be specified and also provides a base which could be extended to provide a more generic descriptive capability for factory scheduling problems.

1 A Domain Description Language for the H1 Model

TOSCA [Beck 93] is an opportunistic scheduling system designed to address job-shop scheduling problems of a realistic scale and constraint complexity.

As part of the TOSCA project undertaken for Hitachi Limited, the Artificial Intelligence Applications Institute at the University of Edinburgh has produced a Domain Description Language for job-shop scheduling. The DDL enables the testbed dataset used in that project (based on real factory data) to be specified and also provides a base which could be extended to provide a more generic descriptive capability for factory scheduling problems.

This document provides the following:

- a description of the generic job-shop scheduling problem,
- a characterisation of the specific nature of the Model H1 problem,
- a detailed description of the components of the Domain Description Language, and
- a syntax for the Domain Description Language.

The full Model H1 is confidential; this document provides an outline of the model.

1.1 Scope of the Document

At present there is no universally accepted scheduling domain description language. The need for a language is however very real, both for the development of a generic scheduling tool and as a basis for the exchange of research ideas and data. This paper is intended to document the features of job shop scheduling domains in order that particular problems and relevant factory descriptions can be specified to future schedule generation systems, such as TOSCA.

The scope of the language should be tested against a range of factory types applying various test data suites. An example of such a test data suite is that published by the scheduling group at the Robotics Institute, Carnegie-Mellon University [Chiang *et al* 89]. Unlike Hitachi's Model H1 dataset, the CMU data is very limited in detail and clearly not based on a real factory scheduling problem, but does provide a useful testbench for a scheduling system and a scheduling input language. The DDL will be driven by the range of factory models to which it is applied; currently, the language is driven primarily by the needs of the Model H1 data.

Given the diversity of manufacturing scheduling, it should be expected that extensions and revisions to the language will be necessary when other models are explored. The language should evolve to provide completeness of description, as far as possible, but in its early form the DDL will, at minimum, act as a feature *checklist*. Difficulty or failure to express some requirement in the language will however provide valuable information on the potential for assessing the descriptive power of a generic formalism.

1.2 Scheduling Elements

At the heart of the description are the following key scheduling elements:

Production: the manufacturing process concerned with the transformation of materials into end-products. Associated with each product is a set of process plans. Each process plan describes a method of production (*i.e.*, a set of temporally ordered operation types).

Demand for Production: imposed by the orders accepted and predicted by the manufacturing system. Demands are descriptions of the obligations for production that the manufacturing system has undertaken. The term *load* is used synonymously in the manufacturing production literature.

Capacity to Produce: the factory resources and production plans. The capacity of the factory resources are described by their capabilities, corresponding to the various operation types which they can process, and their speed of processing.

Production Constraints: conditions which must be satisfied for a schedule to be valid. Overall schedule objectives (*e.g.*, minimise Work-in-Process) are a special type of constraint in that they apply across the entire schedule. Achieving such *global constraints* necessarily involves large numbers of inter-related decisions.

1.3 Distinctive features of the approach to defining the DDL

The DDL is intended to be a language independent of implementation details and scheduler strategies, but in a form which can be readily understood by the user and a scheduling system. The structure of the language will be centred about *descriptive tables*, as would be used in a typical computer database.

In that the prime interest is to allow the language to be used for knowledge-based scheduling, the language should make provision for the full complexity of the factory environment and not rely on simplifying assumptions. Examples of simplifying assumptions commonly adopted in manufacturing resource planning systems include: (i) restricting the search space by not representing alternatives, (ii) ignoring constraints, and (iii) pre-batching orders into higher level production units such as lots.

(i) No representation of alternatives: scheduling systems commonly simplify the resource allocation problem by pre-defining default selections for resources and process plans. Where alternatives do exist they should be represented and selections left to the scheduling system.

(ii) Ignoring constraints: complex constraints (*e.g.*, setup limitations) may be omitted leaving potential scheduling problems to be handled on the shop floor. The knowledge-based scheduling approach aims to model all relevant constraints on production. This also includes the representation of overall scheduling objectives.

(iii) Pre-batching orders into lots: the definition of lots (*i.e.*, setting production quantities, release dates and due dates) significantly defines the production schedule, and limits what can additionally be achieved by the operation schedule. For this reason, the separation of lot definition and the operation scheduling steps should not be assumed.

Problem

- Allocate resources and start times to operations.

Inputs

- Production
- Demand for Production
- Capacity to Produce
- Constraints

Constraints

- Resource capacity
- Temporal relationships between operations

Figure 1: Description of the Generic Job-Shop Scheduling Problem

Also, the approach adopted must make provision for extension and elaboration in the language and implementation.

2 Characterising Job Shop Scheduling as a Generic Problem

The job-shop scheduling problem involves the allocation of resources and start times to a set of operations subject to a number of constraints. The primary constraints handled by Manufacturing Resource Planning[1] systems relate to the demand and capacity of the factory resources and material. In the generation of operation schedules, the availability of material is normally assumed and the focus is on demand for capacity and the capacity constraints.

The goal of the factory is the manufacture of products. The basic scheduling problem is the satisfaction of production requirements given the available factory capacity. The modelling of operation scheduling involves these core elements: (i) production plans, (ii) factory demand and (iii) factory capacity. Figure 1 provides an outline of elements of the generic job-shop scheduling problem.

[1] MRP-II

2.1 Production

Production refers to the definition of processes and materials required to manufacture products. The steps of production for a single product is referred to as a *process plan*. These steps may consist of primitive or compound operations which may or may not be temporally ordered with respect to each other, *i.e.* operations may have a strict ordering (before or after) or may happen in parallel. Process plans are typically represented by networks, or graphs, showing the precedence relationships between substeps. Linear plans are strict sequences of steps.

Process plans may carry overall plan information and requirements, such as estimated processing and production lead times, overall material usage, cost, *etc.* This information may be valuable at a high level of scheduling or schedule analysis, or for making decisions about which process plan, from a possible set of plans, should be chosen as the most suitable production method. Multiple alternative process plans offers additional flexibility to manufacturing planning ad scheduling systems but they introduce significant computational complexity.

2.2 Demand for Production

Demand is imposed by the *orders* accepted or predicted by the factory. These orders (make-to-order or make-to-stock) are for specified quantities of products by a specific date. During the Material Requirements Planning phase of production scheduling, orders may be combined or split for the purposes of production into "units of production" or *lots* — a quantity of items produced together. This step of transforming orders to lots and selecting lot sizes can have a significant impact on inventory costs, setup costs, capacity requirements and scheduling flexibility. The demand for capacity is normally assessed in relation not to the demand for products or for lots, but in terms of the demand for resources required to process the required lots. This involves, for each lot, selecting a process plan and thus defining the set of operations to be performed. Scheduling demand is then calculated on the basis of aggregated operation demand for resources.

Figure 2 shows the relationships between orders, lots, and operations. The lot definition step — the selection of lot sizes, release date and internal due date — is undertaken by the Manufacturing Requirements Planner. The process plan selection step is undertaken by the operation scheduler.

2.3 Capacity to Produce

For most operation scheduling problems, the capacity of the factory is essentially equated with the capacity of each of the individual machines. This assumption regarding the primacy of the machine capacity constraints may not always hold and in such cases it is necessary to explicitly represent other resources (*e.g.*, operators, tools and materials).

Figure 3 shows the various factory resources to be managed during factory production. Operation scheduling systems typically omit many of these entities from the factory model and focus essentially on the most constrained resources, normally machines. Job-shops normally view the factory in terms of work centre responsibilities (*i.e.*, production and accounting) and for this reason it may be necessary to represent various resource

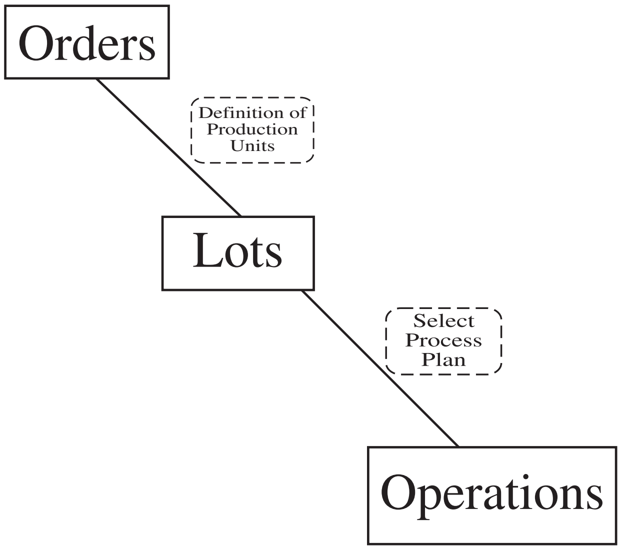

Figure 2: Decomposition of Factory Demand

groupings. The organisation of resources (*i.e.*, their physical layout) usually imposes additional capacity constraints and for this reason are treated as separate capacity entity type.

2.4 Constraints

It is the task of the scheduler to assign all the operations of all the lots to suitable resources for processing, subject to a variety of constraints such as: (i) meeting due dates; (ii) observing capacity constraints on resources; and (iii) restrictions on resource changeovers, or setups.

This list, the major constraints of the H1 problem, is by no means exhaustive. Other constraints will be present in other production environments (*e.g.*, legal constraints on work practices, use of overtime, *etc.*)

Objectives: The overall factory objectives of the scheduler may be viewed as constraints of particular significance in that they affect the *overall scheduling strategy* (*i.e.*, they influence decisions throughout the schedule generation process.)

For any specific problem instance there are generally very many possible schedules which could be generated, though not all are equally desirable. The desirability or quality of a solution may be assessed in various ways:

(i) a set of independent assessments based on a single feature of the schedule (*e.g.*, total number of lots scheduled, makespan *etc.*).

(ii) conformance to a set of requirements (*e.g.*, scheduling more than 80% of the lots)

Resource Types

- Machines
- Operators
- Tools
- Transport
- Inventory
- Storage Areas
- Work Centres

Figure 3: Types of Factory Resources

(iii) an objective function which provides a single integrated assessment of the schedule quality (*e.g.*, a function of lots scheduled and makespan).

These various metrics are used to assess the quality of the generated schedule and may also be used to inform and direct the schedule generation process itself. This is the approach used in both the ISIS [Fox & Smith 84] and OPIS [Smith 87] scheduling systems. An alternative approach to achieving the overall objectives is to define *strategies* for achieving the result. Scheduling strategies may indicate local decision preferences which correspond to a particular global scheduling objective. A JIT scheduling strategy applied to a job-shop, for example, would indicate a preference for operations to start as late as possible without making lot production late. This has the effect of reducing the overall Work-In-Process in the factory [Berry 91, Sadeh 91].

3 Distinguishing characteristics of the Model H1 Job Shop Scheduling Problem

Before defining the Domain Description Language it is instructive to comment on the generality, complexity and limitations of the H1 model. H1 is intended to be a realistic subset of the general D class of job shop problem, and therefore covers a rich variety of deterministic machine scheduling problems.

The problem class is sufficiently realistic to ensure that answering the simple question of whether a feasible solution exists involving all lots is far from trivial.

In that Model H1 is based on a specific job-shop, it is expected that its definition cannot be treated as generic. The following features are not explicitly included in the H1 model, though it is desirable that a DDL should permit their possible incorporation.

3.1 Production

3.1.1 Products

A characteristic of the job-shop is the intermittent manufacture of relatively large numbers of products. Model H1 is an example of a highly-flexible production environment capable of manufacturing more than 180 different products. A concomitant feature of such an environment is the requirement for relatively frequent machine changeovers.

3.1.2 Temporal orderings of operations within process plans

The process plans of Model H1 defines sets of operations which are fully temporally ordered; that is, operations occur in sequence one after the other. Assembly lines do impose this strict temporal ordering on operations but it is frequently the case that, in the job-shop setting, operations are only partially ordered allowing for potential concurrency of processing.

3.1.3 Alternative process plans

Model H1 assumes that there is only a single process plan for each lot type, *i.e.* there is no choice of plan. This is often not the case as there can either be:

1. different process plans for a lot, or
2. choice of operation within a single process plan.

Either way there can be a range of options from which a choice must be made. In the second case, the exponential branching of process plan options is implicit.

3.2 Demand for Production

3.2.1 Order details

Priorities on orders and lots: In the H1 model there are no explicit customer, or other, priorities associated with a particular order or lot. Priorities on orders can have a variety of possible derivations, such as customer or cost associated with the processing of the lot. Dispatching heuristics applied for operation sequencing frequently incorporate such static priorities.

3.2.2 Lot preferences

Temporal preferences: Lot preferences may be assigned on the basis of overall factory scheduling objectives or alternatively from a preference associated with a particular factory order. An overall factory scheduling objective such as JIT can be translated as a lot preference to: schedule each operation as late as possible without violating the lot's due date constraint. Some factory lots may be "rush orders", a condition reflecting a lot preference to schedule each operation as early as possible. The handling of multiple conflicting scheduling objectives is a feature of the required flexibility of Model H1.

3.2.3 Operation durations

Time specifications involving uncertainty: Operation durations for H1 involves (i) a processing duration and (ii) a setup duration. The processing durations of operations are *fixed but conditional on the resources used for their processing.* In the case of Axial machines, processing duration is affected by having or not having a walking beam capability. Those Axial machines without walking beams are limited to a minimum processing rate of 1 unit in 0.3 minutes; Axial machines with walking beams are not subject to this restriction being able to process some types of operation considerably faster.

The duration of setups are also fixed but not always required. Where the previous operation is of the same type and process plan (*i.e.*, same process plan operation (*c.f.*, Section 4.1.3), no setup is needed (*i.e.* the setup duration is zero) and the operation duration is just the processing duration.

3.2.4 Operation preferences

Resource preferences: Operations may have specific resourcing preferences. One machine may be particularly reliable or cost-effective and for this reason is always the first choice machine. Alternatively, resourcing preferences may be more specific being associated with certain products (or process plans) only. The Model H1 dataset does not indicate resourcing preferences.

Temporal preferences: Temporal preferences which apply to operations are inherited from the preference associated with the processing of a lot. Regarding the assignment of preferences, refer to Section 3.2.2 above.

3.3 Capacity to Produce

3.3.1 Machines

Machine capacity: A machine's capacity is described by (i) its capabilities (*i.e.*, the range of operation types it can process), (ii) its speed of processing, (iii) the time taken to setup and (iv) its availability. In Model H1, there are four types of machines: Universal Jumper, Axial, Radial and Integrated Circuit machines. Within these machine types, the individual machines have *identical*, *overlapping* or *disjoint* capabilities. Two machines have

- identical capability, if the sets of operation type which they can process are identical (*i.e.*, all operations in common);
- disjoint capability, if the sets of operation types which they can process are disjoint (*i.e.*, no operations in common);
- overlapping capability, if the sets of operation types which they can process are overlapping (*i.e.*, some operation or operations in common).

The processing speed is product-specific and the setup time depends on whether a changeover is indeed needed. Machine availability is normally assumed for all times other than when preventive maintenance in planned. To allow consideration of unexpected downtimes, mean time between machine failures might be represented. This information is not provided as part of the H1 dataset.

3.3.2 Operators

Operator constraints: The requirement for human involvement during production — to setup and monitor machines — places additional constraints on the schedule. These *operator constraints* are usually treated as secondary constraints but may be of significant importance where operator skills are scarce. In such cases, operator constraints may be treated as additional primary constraints. In Model H1, although operator constraints are not explicitly provided, it is assumed that these constraints are present and indirectly represented by the setup constraints.

3.3.3 Factory layout

Work centres: The layout of the factory affects capacity in various ways. Work areas may impact on inventory, transportation, operator and routing constraints. In Model H1, inventory and transportation data is not provided and no restrictions on production routings are imposed.

3.3.4 Setups

Setup times functionally related to transition states: Setup durations in Model H1 are defined by machine type. Integrated Circuit machines take 60 minutes for a changeover, Universal Jumper machines take 1-2 minutes, Axial machines take 10 minutes, and Radial machines take 25 or 35 minutes depending on whether a load magazine change is also involved. A change in the pitch setting of an Axial machine is possible, but because it is a particularly lengthy activity (450 minutes), this setup is assumed not to occur.

Setup durations in some contexts are dependent not just on machine type but also on the specific product (operation type) change. To represent such changeovers, two-dimensional setup duration matrices are needed.

Setup constraints: Three types of setup constraints are defined in Model H1. Two of these constraints pertain to the rate of setups permitted[2]: (i) for individual machines and (ii) for groups of machines (work centres). The third constraint pertains to the duration bounds between setups for a group of machines or work centre.

The justification for these constraints is not made clear in the problem and may include both operator constraints and also factory scheduling heuristics. Scheduling heuristics should, where possible, be distinguished from the problem specification.

[2]The problem specifies a maximum number of setups for a given work period such as a shift.

4 Components of the Domain Description Language

The Domain Description Language aims to represent detailed descriptions of a generic factory production model. In addition, it is intended to offer a clear and readily understandable model and description of a particular problem.

The factory is viewed as comprising three primary categories, namely:

1. Production;
2. Demand for Production;
3. Capacity to Produce;

Figure 4 shows these categories and the major entities which fall within these categories.

This section covers each of these categories and their entities. Each entity is explained below via tables of components similar to standard relational databases. Examples are given in order to highlight the various relationships.

4.1 Production

Production is defined by the following entities: Products, Process Plans and Process Plan Operations. The products of the factory include the intermediate products as well as the final customer product. Associated with each product is one or more method of production (*i.e.*, one or more process plans), and associated with each process plan is a set of steps (*i.e.* types of operation).

4.1.1 Products:

Product		
Product Id	Product Description	Level

Product Id: A label to uniquely identify the product.

Product Description: A textual description of the product.

Level: The level indicates the position of the item or product within the Bill of Materials. Level 0 refers to the final product, level 1 refers to the components of the final product. Figure 6 showing the component decomposition of products indicates the increasing level of end-products through to intermediate-products.

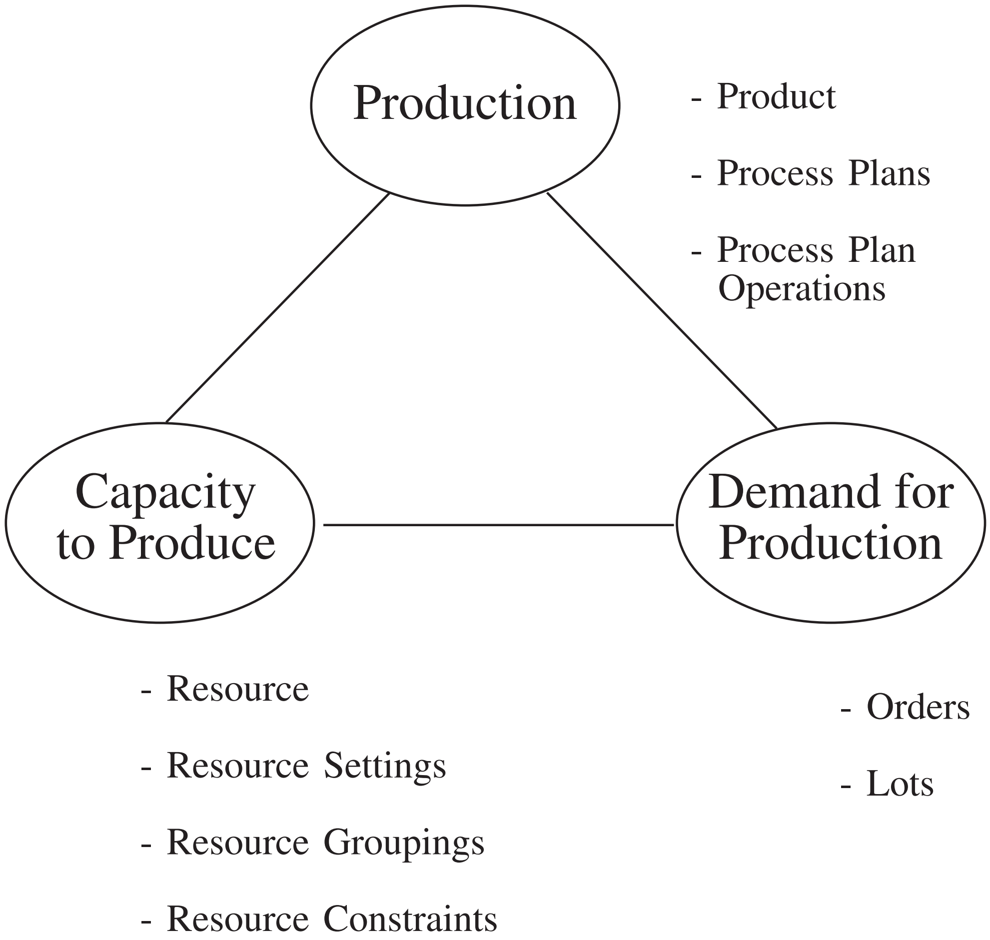

Figure 4: Overview of the Elements of the DDL

Demand

- Products
- Process Plans
- Process Plan Operations

Figure 5: Entities Defining Job-Shop Production

4.1.2 Process Plans:

Process Plans				
Process Plan Id	Product Id	Lead Time (estimate)	Material Cost	Ideal Lot Size

Information for the process plans is restricted to aggregated measures, such as lead time and material use. The operation details of a plan can be readily derived from the operations table below, from which orderings, *etc.* can be determined.

Process Plan Id: a label to uniquely identify the process plan.

Product Id: a pointer to the product manufactured by this process plan.

Lead Time Estimate: an estimated value of production lead times if this plan were to be used. This figure scaled by the quantity can be used to estimate lead time for a lot. This value may be used to choose one plan from a set of possible applicable plans. The nominal lead time can also be used for the purpose of comparison with scheduled lead times.

Material cost: The aggregated cost of all materials. This factor can influence scheduling strategy (*e.g.*, a scheduling strategy to minimise inventory is particularly important for production lots with costly materials.)

Ideal Lot Size: The ideal production lot size for this product type and process plan. Whatever lot size is selected, there are inevitably implicit advantages and disadvantages. Larger lot sizes reduce the need for setups and reduce the complexity of shop floor monitoring; smaller lot sizes are more easily scheduled and allow lower inventory levels to be maintained. The 'ideal lot size' defines the optimal balance. Lot size is often constrained by economy of scale related to material purchase.

4.1.3 Process Plan Operations:

Process Plan Operations							
PP-Op Id	Resource Setting Id	Process Plan Id	Process Duration	Material Requirement	Previous Op.	Next Op.	Delay Bounds

Each operation in this table is individually described in relation to other operations in the process plan. By doing so the *orderings* within the process plan are made explicit, thereby enabling various process plan topologies to be represented, *e.g.* in-tree plans, linear plans and graphs. An important by-product of this representation is that it matches well with the representation of the setups operations enabling simple lookup of setup times, *etc.* This neatly separates the various information tables.

Figure 7 shows a partially-ordered and fully-ordered process plans. In the case of the partially-ordered plan the implicit parallelism between operations op2, op3 and op4 is retained.

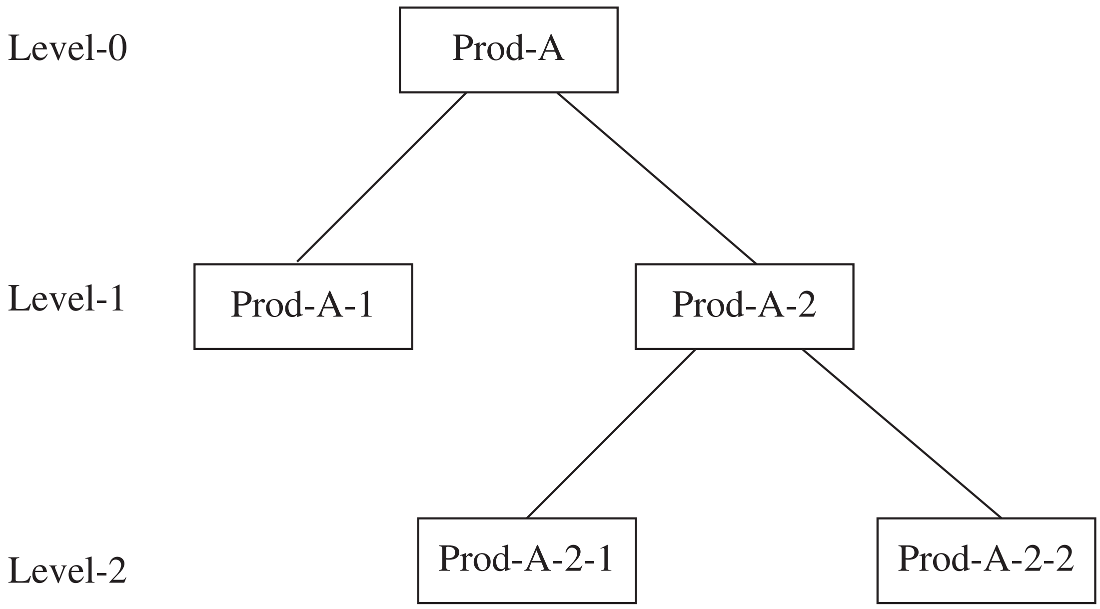

Figure 6: Product Component Decomposition

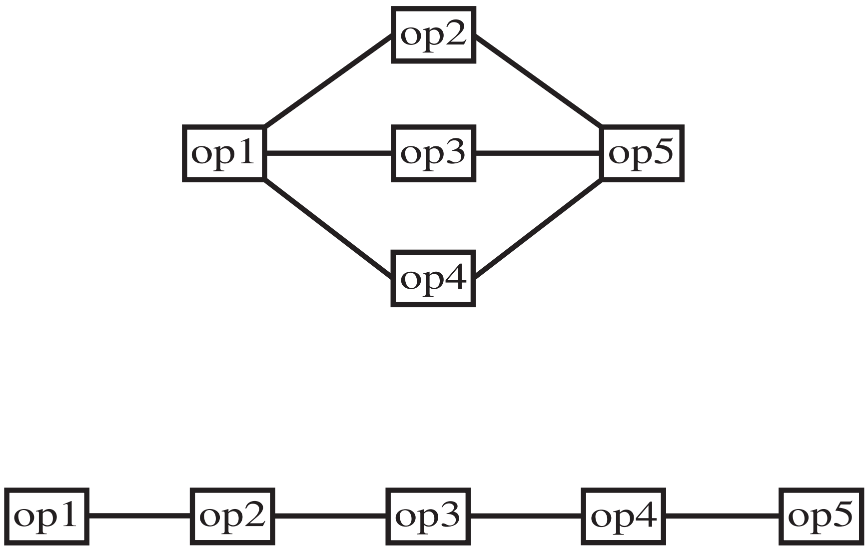

Figure 7: Partially- and Fully-Ordered Process Plans

PP Op Id: a label to uniquely identify the operation in a specific process plan, *e.g.* `op-1-plan-part-X`.

Resource Setting: a pointer to the required resource setting.

Process Plan Id: a pointer to the process plan containing this operation.

Processing duration: the time to complete the processing of a single unit. The processing duration of a lot is the processing duration of an operation multiplied by the lot size (*c.f.* setup table for the definition of setup duration).

Materials Required: A listing of the components (subassemblies and raw) materials needed for the processing of this process plan operation. This field corresponds to a single level decomposition of a sub-assembly's *Bill of Materials*, sub-components (*i.e.*, the components of the components) need not be included. Figure 6 shows the compositional relationships between products.

Pre-Op: a pointer to the predecessor operation in the specific process plan. In the case of a non-linear process plan this value could be a *list* of operations.

Next-Op: a pointer to the successor operation in the specific process plan. In the case of a non-linear process plan this value could be a *list* of operations.

Delay Bounds: operations do not necessarily follow on immediately in time from the previous operation. There are many reasons for introducing a deliberate delay between operations, *e.g.* to allow a component to cool after heating. This delay can have both a lower and an upper limit, captured in the time bounds specified.

4.2 Demand for Production

Demand is defined by the following entities: Orders and Lots. For an integrated planning and scheduling system the specification of factory lots would not be required. Currently, however, lots (with size, date of release and due date specified) are normally provided to the operation scheduler. An intermediate entity, a Sub-order, is defined to allow the many-to-many relationship between orders and lots to be represented.

Operation instances are generated during the scheduling process and not included in the DDL. Detailed operation demand is calculated by the operation scheduler following the selection of a process plan and the creation of time-restricted operations.

4.2.1 Orders

Orders						
Order Id	Product Id	Customer	Quantity	Priority	Due Date	Tardiness Costs

An order, distinguished by order id, refers to a single product order from one customer. Where a customer orders a set of products, separate orders will be created for each product.

Demand
• Order • Lot

Figure 8: Entities Defining Job-Shop Demand

Order Id: a label to uniquely identify the order, *e.g.* `order-1234`.

Product Id: pointer to the item to be manufactured.

Customer: a text string identifying the final customer.

Quantity: the number of products placed on order.

Priority: different customers and orders will be allocated a different processing priority level, depending on factors such as value of order and relationship with customer. The absence of an explicit priority indicates a defaulted priority dictated by the processing strategy of the individual lots.

Due Date: a time specification at which the order should be completed. This does not necessarily correspond to the production due dates.

Tardiness Costs: additional costs and penalties may be associated with order tardiness. This can be used to select scheduling strategy and lot priority.

4.2.2 Sub-orders:

Sub-orders			
Sub-order Id	Order Id	Lot Id	Quantity

Lots form the production batches defined by the factory. Product orders are subdivided into lots either in a one-to-one or one-to-many relation if they originate from a single production order entry, or in a many-to-many relation if they represent an aggregation of many product orders. Sub-orders, shown in Figure 9, are used to represent the (possibly) many-to-many relationship between orders and lots.

Demand may be specified by order or by lots, the latter being the norm for the H1 environment. Normally only one will be used. However, if a mixture of lots and orders are given, then any relationship between them must be specified via sub-orders, otherwise the individual lots and orders are interpreted as being additional to one another.

Sub-order Id: a label to uniquely identify the sub-order.

Order Id: a pointer to the order of which the sub-order is a part. (The sub-order could of course be the entire order.)

Figure 9: Sub-orders linking Orders and Lots

Lot Id: a pointer to the lot of which the sub-order is a part. (The sub-order could of course be the entire lot.)

Quantity: the number of items in the sub-order.

4.2.3 Lots:

Lots						
Lot Id	Process Plan Id	Processing Strategy	Product Id	Quantity	Nominal Lead Time	Material Cost

Lot Id: a label identifying the lot, *e.g.* `lot-99`.

Preferred Process Plan Id: an optional pointer to the preferred process plan for this lot.

Processing Strategy: there will be a number of possible processing strategies for a lot, *e.g.* (i) Critical or Rush orders which require urgent processing, (ii) Make-to-Order items produced in a JIT fashion, or (iii) Make-to-Stock Orders which can be processed in any fashion within the overall objectives of the factory.

Product Id: a pointer to a specific product.

Quantity: the size of the lot, *e.g.* 20 items.

Material Cost: the cost of raw materials consumed in the production of the lot.

4.3 Capacity to Produce

The capacity to produce is defined by the following entities: Resources, Resource Groupings, Resource Capabilities and Resource Constraints.

Whereas resources may be viewed as the primary entity type related to production, it is also important to distinguish Resource Groupings which are subject to *additional constraints* (*e.g.*, maximum rate of setups at a workcentre). Separate tables are provided to represent Resource Constraints and Resource Capabilities (*i.e.*, the range of factory capabilities.)

Capacity
• Resources • Resource Groupings • Resource Capabilities • Resource Constraints

Figure 10: Entities Defining Job-Shop Capacity

4.3.1 Resources:

Resources are the physical entities required to perform manufacturing activities. They include: machines, operators, tools and inventory. The complexity of the model of a manufacturing environment used by operation schedulers is normally restricted to *primary resources*[3] only (*i.e.*, the resources which most constrain production.) OPIS defines *primary resources* to be stationary resources, those with a fixed location within the factory. Mobile resources are generally treated as secondary resources, but OPIS does make provision for defining human operators as being the primary resource too [Smith 89].

Resources			
Resource Id	Description	Availability	Settings
Resource Type	Mean time to Failure	Repair Duration	Demonstrated Capacity

Resource Id: a label to uniquely identify the individual resource.

Description: a short textual description of the resource.

Availability: resources may have a limitation on their overall availability, or be available for certain periods only. This is distinguished from non-availability associated with preventive maintenance (*c.f.*, Section 4.3.4).

Settings: the range of possible resource settings (*i.e.*, operation types which this resource can perform).

Resource Type: describes the usage or production characteristics of the resource. The following characteristics have been identified:

consumable_strictly: A set amount of the resource is available and cannot be topped up.

[3] Primary resources are normally the most expensive factory items.

consumable_producable_by-factory: The resource can be topped up from scheduled operations within the factory under the control of the scheduler.

consumable_producable_outwith-factory: This resource is similar to consumable_producable_by-factory except that extra resource is only obtained via an off-line process such as a delivery rather than via scheduled operations within the factory under the control of the scheduler.

consumable_producable_by-and-outwith-factory: This resource is a combination of the two above. Resources can be produced by both factory actions and by off-line processes.

reusable_non-sharable: The resource is allocated from a "central pool" in unit amounts and when the resource is finished being used it is then deallocated back to the pool; *e.g.*, workmen, robots, lorries.

re-usable_sharable_independently: The resource can be shared without coordination to specific time periods, *e.g.*, space in a holding area.

sharable_synchronised: The resource is shared for a specific time, *e.g.* capacity on a particular journey of an AGV.

Mean Time to Failure: a measure of machine reliability.

Repair Time: a measure of the disruption to be expected by a machine failure.

Demonstrated Capacity: the proven capacity calculated from actual (historical) output performance data. It is normally represented as the average number of items produced per hour.

4.3.2 Resource groupings:

Factories define specific production facilities or work centres which can be considered as a single unit for the purposes of capacity requirements analysis. Resource groupings usually have additional constraints, not associated with individual resources (*e.g.*, available operators, tools and inventory.)

Resource Grouping		
Resource Group Id	Description	Resources

Resource Group Id: is a simple label, such as `Lathes`, which simply identifies a set of machines.

Description: is a short textual of the family of resources, *e.g.* `Manual Milling Machines`.

Resources: the set of resource ids which constitute the resource group.

4.3.3 Resource capabilities:

Resource capabilities are represented by two table: one describing the functional capabilities (*i.e.*, the various resource *settings*), and the other describing the *transitions* from one setting to another.

(a) Settings: The potential settings of a resource characterises its functionality. Each different setting (or setting range) defines the types of operations which the resource can process.

Settings				
Setting Id	Resources	Processing Speed	Operator Required	Tools Required

Setting Id: a unique description of the setting.

Resources: the set of resources which are capable of such a setting.

Processing Speed: the time taken to process a single item.

Operator Required: the skill requirement of the operator to use the resource at a particular setting.

Tools Required: the set of tools required for a particular setting.

(b) Setups: The duration of the changeover from one setting to another is defined by the setups table.

SetUps					
Setup Id	Resource Id	From Setting	To Setting	Duration	Operator Required

Setups are performed on a resource as it changes from one functional state (*i.e.*, resource setting) to another. This table essentially specifies a set of transition tables for each resource, indicating the time and resources required to put into effect the setup.

Setup Id: a unique description of the specific setup.

Resource Id: a pointer to a specific resource, or resource family if setup information is common throughout.

From/To Settings: setup times may be a function purely of the required functionality (the `To Setting`), in which case this need be the only resource setting specified. However other setups may be a function of both the current functionality *and* the intended functionality, *i.e.* the times are a variable which can only be determined during schedule generation.

Duration: the time to put the setup into effect.

Operator Required: the skill requirements of the operator to changeover to this particular setting.

4.3.4 Resource constraints

A wide range of possible constraint types may apply within a factory. In the DDL, three constraint types are defined:

(a) Non-availability of resources

(b) Processing quantity constraints

(c) Setup constraints

(a) Resource Non-Availability

Information regarding resource non-availability as a result of planned preventive maintenance must be input to the operation scheduler. Preventive maintenance may be pre-scheduled and treated as given; alternatively, maintenance may simply be treated as special operations to be allocated by the scheduling system.

Non-Availability			
Resource Id	From Time	To Time	Reason

Resource Id: a pointer to a specific resource.

From/To Time: time point specifications delimiting the non-availability of the resource.

Reason: a textual description of why the resource is unavailable, *e.g.* down for preventive maintenance.

(b) Processing quantity constraints

Resources have limited processing capacities which must be adhered to by the planning and scheduling system. For instance, an oven may allow only 100 items to be heated at one time. These constraints may or may not be fully handled by the planning system in choosing lot sizes.

Processing Quantity Constraint			
PQC Id	Resource Id	Minimum Quantity	Maximum Quantity

Processing Quantity Constraint (PQC) Id: a unique identifier for a processing quantity constraint[4].

Resource Id: a pointer to a specific resource.

Minimum Quantity: the minimum number of items which can be processed in a single batch at a resource. This is most likely to be imposed by the factory.

Maximum Quantity: the maximum number of items which can be processed in a single batch at a resource

[4]As far as the DDL is concerned, ids must be presented. It is expected that ids such as the Processing Quantity Constraint Id would be automatically generated.

(c) Setup constraints

The management of setups is an important determinant of the quality of the final schedule. The management of setups refers primarily to the overall number of setups, but the task is not simply to minimise setups. Setups do involve an overhead limiting overall productive utilisation but also provide the necessary flexibility for a job-shop to manufacture multiple products to time.

The number of setups, for a given product mix, is very largely set by the size of lots selected, but can also be affected by the sequencing of lots by the operation scheduler.

Setup constraints may reflect the desire to restrict the total number of setups or alternatively may simply reflect physical constraints on what is achievable, for example by one or two human operators. These two setup constraint types (*maximum rate* and *non-overlapping*) are described below.

(i) Maximum rate

The overall number of setups may be restricted by a factory by placing an upper limit on the permitted setup rate. For example, the factory may impose a limit of 3 setups per shift on a single machine or 10 setups per shift on a group of machines.

Max rate of setups			
MRS Id	Resource Id	Max num Setups	Timeperiod

Maximum Rate Setup (MRS) Constraint Id: a unique identifier of a maximum rate setup constraint.

Resource (or Resource Group) Id: a pointer to a resource or resource group.

Time Period: a time period used to define the permitted setup rate for the resource or resource group.

Maximum Number of Setups: the maximum number of setups which are permitted to occur within the specified time period.

(ii) Non-Overlapping Setups

A human operator is usually required throughout the setting up processing for a resource. For this reason, overlapping setups is restricted by the number of available operators. Where only one operator is available for a group of machines, only one setup process can be active at any one time; where N operators are available for a group of machines only N setup processes can be active at any one time.

Non-overlapping setups			
NOS Id	Res Group Id	Enforced time Between Setups	Num Overlapping Setups Permitted

Environment Variables

- Scheduling Horizon
- Schedule Granularity
- Schedule Rating
- Schedule Strategy
- Time Unit of Schedule
- Cost Unit of Schedule
- Time Unit Conversions

Figure 11: Environment Variable

Non-Overlapping Setups (NOS) Constraint Id: a unique identifier of a non-overlapping setups constraint.

Resource Group Id: a pointer to a resource group.

Enforced Time Between Setups: the time required for a setup. This may be a conservative estimate of the time needed.

Number Overlapping Setups Permitted: the number of setups that can be active at one time.

4.4 Environment Variables

The environment variables are global measures defining quantities and values within which the schedule must operate. The variables defined are: the scheduling horizon, the schedule granularity, schedule function, scheduling strategy, unit of the schedule, unit conversions.

Scheduling Horizon: a period over which the schedule is to be generated, *e.g.* a list of days.

Schedule Granularity: a temporal granularity to which the schedule (*i.e.* operation start times) must be specified. This may, for example, restrict start times to the nearest 10 minutes.

Schedule Rating: a function used as an index of the overall quality of the final schedule. It will in general be a composite of a number of individual ratings and their weights.

Schedule Strategy: a high-level (generic) factory strategy used to direct the scheduling process (*e.g.* JIT or minimal cost). The extension of the range of permitted strategies (*e.g.* to include maximal robustness) depends on current and future research.

Time Unit of Schedule: the smallest time unit used for the specification of the schedule (*e.g.* minute).

Cost Unit of Schedule: the currency used to define costs in the schedule.

Time Unit Conversions: a set of larger scale time units in terms of the smallest unit of the schedule; *e.g.*, ((hr 60)(shift 480)).

5 DDL Syntax

The syntax presented seeks to provide a detailed representation of job-shop scheduling problems. It should be expected that the form will be elaborated and modified as more detail becomes available.

The syntax adopted in the TOSCA DDL closely follows that adopted in the *Task Formalism* (TF) domain description language used in Nonlin [Tate 77] and O-Plan [Currie & Tate 91, Tate *et al* 93].

5.1 Conventions Used in the DDL Description

Some simple conventions used throughout deserve explanation before the DDL language is defined:

Keywords Keywords used in DDL are written in **bold** lettering for highlighting purposes in the descriptions of statements.

Definitions DDL components are surrounded by angle brackets “< >”.

Options Optional words or phrases are surrounded by square brackets “[ ]”.

Choice If there is more than one possible representation for an expression then second and subsequent definitions are separated by the vertical bar character “|”.

Repetition If a structure can be repeated indefinitely then this is indicated by three dots “...” occurring directly under (i.e. aligned with) the structure to be repeated or on the same line where the meaning should be clear.

Component Definition Following the definition of a statement, a number of components may be further defined. This is indicated by following the name of the component by “::=” and its definition.

Punctuation Use is made of two punctuation marks in DDL statement definitions, namely

1. “;” which indicates the end of a statement type (for example a condition list).

2. “,” which is used as a list separator within statements.

Neither is optional and their omission would cause a compile error.

Comments Comments may be included anywhere in DDL descriptions and are introduced by three semicolons “;;;” Everything following these in the line is treated as comment and would be ignored by a DDL Compiler. .

5.2 Component Definitions

There are a number of components of DDL forms which are used in various places. These are defined here.

5.2.1 Basics

The following defines some basic components used in many DDL forms.

```
<name> ::= <letter> [ <letter> | <digit> | _  | % ]
                    ...

<number> ::= <integer> | <float> | inf | infinity
                     inf = infinity = a number larger than any other

<float> ::= [ <sign> ] <digits> [. <digits> ] [ <exponent> ]

<digits> ::= <digit> ...

<digit> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<text_string> ::= " <character> ... "
```

5.2.2 Time Specifications

```
<time_spec> ::=    <time_units>
                 | <hours> : <minutes> [ : <seconds> ]
                 | <days> ~ [ <shifts> ~ ] <hours> : <minutes>
                                                  [ : <seconds>]

<days> | <shifts> | <hours> | <minutes> | <seconds> ::= <integer>

<time_units> ::=    seconds
                 |  minutes
                 |  hours
                 |  shifts
                 |  days
```

```
<time_bounds_spec> ::=  <time_bounds_pair> [ with <time_preference> ]
                     |  ( <time_bounds_pair> [ with <time_preference> ] )

<time_bounds_pair> ::=  [ occurs_at ] <time_spec>
                     |  [ et = ]       <time_spec> [ .. ]
                        [ lt = ]       <time_spec>
                     |  [ between ] [ et = ] <time_spec>
                        [ and | .. ] [ lt = ] <time_spec>
                     |  after <time_spec>
                     |  > <time_spec>
                     |  before <time_spec>
                     |  < <time_spec>

<time_preference> ::=  ideal = <time_spec>
                                 other preferences are being considered
```

The specifications of <days>, <hours>, etc. are integer numbers. All time specifications map to a number of time units relative to some absolute zero time: 0~0~0:0:0. In the absence of the explicit specification of the <time_units> seconds is the default. The special symbols **inf** and **infinity** may be used and map to a number larger than any other number.

All time window specifications map onto the same form as shown in the table below.

	est/min	lst/max
occurs_at t	t	t
t	t	t
et=t1..lt=t2	t1	t2
t1..t2	t1	t2
et=t1..lt=t2	t1	t2
between t1 and t2	t1	t2
after t	t	infinity
> t	t	infinity
before t	0	t
< t	0	t
default window	0	infinity

In fact the **initial_time** specification for the plan will serve to improve the lower bound specifications when the planner operates.

5.2.3 Cost Specifications

Costs are given with respect to some monetary currency. Cost penalties are currently defined as a single cost associated with a delay; this may need to be elaborated to permit penalty cost functions.

```
<cost_spec>  ::=    <number> [<cost_units>]
```

```
<cost_units> ::=    pounds | dollars | yen | ...

<penalty_cost_spec> ::= <cost_spec> [after] <time_spec>
```

5.2.4 Unit conversion

Time conversions are provided between the smallest granularity time unit used in the schedule and all other defined time units.

```
<time_unit_conversion> ::= ( <time_unit> <number> )
```

5.2.5 Sets

Sets are surrounded by parentheses "(" and ")". They may be sets of names, or may be sets of more general items which includes numbers, patterns, etc.

```
<name_set> ::= ( <name> ... )

<time_bounds_spec_set> ::= ( <time_bounds_spec> ... )

<rating_set> ::= ( <rating> ... )

<time_unit_conversion_set> ::=  (  <time_unit_conversion> ... )

<other_set> ::= ( <other_member_to_be_defined> ... )
```

5.2.6 Rating Specifications

A rating specification is a set of weighted rating units. The set of rating units may need to be extended to allow other scheduling statistics to be incorporated. Some of these may be problem-specific.

```
<rating_spec> ::=   <rating_set>

<rating>      ::=   <weighting> [<rating_unit>]

<weighting>   ::=   <number>

<rating_unit> ::=   percentage_lots_allocated |
                    mean_lot_duration |
                    mean_resource_utilisation   ;;; extensions expected
```

5.3 DDL Forms

Domain information is provided to a scheduling system via a DDL input file which would be translated by a DDL compiler. A DDL file is made up of DDL forms. The DDL language syntax is designed to allow a compiler to operate in a single pass over the input file.

```
<ddl_file> ::= [ <ddl_form> ]
                 ...
```

There are a limited number of DDL forms. They can be given in any order, and more than one copy of a particular form can appear in any separate <ddl_file>. Later forms add to the information extracted from earlier forms. The only requirement on the order in which forms are given, is that information used in later forms is available before use.

```
<ddl_form_keyword> ::= products | process_plans |
                       process_plan_operations | orders |
                       suborders | lots | resources |
                       resource_groupings | resource_settings |
                       resource_non_availability |
                       process_quantity_constraints |
                       maximum_rate_of_setups |
                       non_overlapping_setups |
                       environmental_variables
```

DDL forms have a regular structure. Each is introduced by a keyword and ends with a semi-colon. Internal terms are separated by ",". Each form is introduced by a keyword and ended by **end_<keyword>**. Thus the general structure of DDL is:

```
<ddl_form> ::=  <ddl_form_keyword>
                    <component> ,
                      ...
                      ...
                end_<ddl_form_keyword>
```

5.3.1 Production

- **Products**

```
products
  <product_id>, <product_description>, <level>;
  ...
end_products;
```

```
<product_id>::=<name>
<product_description>::=<text_string>
<level>::=<digit>
```

- **Process Plans**

```
process_plans
  <process_plan_id>, <product_id>, <lead_time_est>, <material_cost>,
  <ideal_lot_size>;
  ...
end_process_plans;
```

```
<process_plan_id>::=<name>
<product_id>::=<name>
<lead_time_est>::=<time_bounds_spec>
<material_cost>::=<cost_spec>
<ideal_lot_size>::=<digits>
```

- **Process Plan Operations**

```
process_plan_operations
  <pp_op_id>, <resource_setting>, <pp_id>, <process_duration>,
     <material_requirements>, <previous_op>, <next_op>, <delay_bounds>;
  ...
end_process_plan_operations;

  <pp_op_id>::=<name>
  <resource_setting>::=<name>
  <pp_id>::=<name>
  <process_duration>::=<time_bounds_spec>
  <material_requirements>::=<name_set>
  <previous_op>::=<name_set>
  <next_op>::=<name_set>
  <delay_bounds>::=<time_bound_spec>
```

5.3.2 Demand for Production

- **Orders**

```
orders
  <order_id>, <product_id>, <customer>, <quantity>, <priority>,
     <due_date>, <tardiness_cost>;
  ...
end_orders;

  <order_id>::=<name>
  <product_id>::=<name>
  <customer>::=<text_string>
  <quantity>::=<digits>
  <priority>::=<name>
  <due_date>::=<time_spec>
  <tardiness_cost>::=<penalty_cost_spec>
```

- **Suborders**

```
suborders
  <suborders_id>, <order_id>, <lot_id>, <quantity>;
  ...
end_suborders;

  <suborders_id>::=<name>
  <order_id>::=<name>
  <lot_id>::=<name>
  <quantity>::=<digits>
```

- **Lots**

```
lots
   <lot_id>, <preferred_pp_id>, <processing_strategy>, <product_id>
      <quantity>, <nominal_lead_time>, <material_cost>;
   ...
end_lots;

  <lot_id>::=<name>
  <preferred_pp_id>::=<name>
  <processing_strategy>::=<name>
  <product_id>::=<name>
  <quantity>::=<digits>
  <nominal_lead_time>::=<time_bounds_spec>
  <material_cost>::=<cost_spec>
```

5.3.3 Capacity to Produce

- **Resources**

```
resources
   <resource_id>, <resource_description>, <resource_availability>,
      <resource_settings>, <resource_type>, <mean_time_to_failure>,
      <repair_duration>, <demonstrated_capacity>
   ...
end_resources;

  <resource_id>::=<name>
  <resource_description>::=<text_string>
  <resource_availability>::=<time_bounds_spec_set>
  <resource_settings>::=<name_set>
  <resource_type>::=<name>
  <mean_time_to_failure>::=<time_bounds_spec>
  <repair_duration>::=<time_bounds_spec>
  <demonstrated_capacity>::=<digits>
```

- **Resource Groupings**

```
resource_groupings
   <resource_group_id>, <resource_description>, <resources>;
    ...
end_resource_groupings;

  <resource_group_id>::=<name>
  <resource_description>::=<text_string>
  <resources>::=<name_set>
```

- **Resource Settings**

```
resource_settings
  <resource_settings_id>, <resources>, <processing_speed>,
     <operator_required>, <tools_required>;
  ...
end_resource_settings;

  <resource_settings_id>::=<name>
  <resources>::=<name_set>
  <processing_speed>::=<time_bounds_spec>
  <operator_required>::=<name>
  <tools_required>::=<name>
```

- **Resource Setups**

```
resource_setups
  <resource_setup_id>, <resource_id>, <from_setting>,
     <to_setting>, <duration>, <mean_time_to_fail>,
     <repair_duration>, <demonstrated_capacity>;
  ...
end_resource_setups;

  <resource_setup_id>::=<name>
  <resource_id>::=<name>
  <from_setting>::=<name>
  <to_setting>::=<name>
  <duration>::=<time_bounds_spec>
  <operator_required>::=<name>
```

- **Resource Non-Availability**

```
resource_non_availability
  <resource_id>, <from_time>, <to_time>, <reason>;
  ...
end_resource_non_availability;

  <resource_id>::=<name>
  <from_time>::=<time_bounds_spec>
  <to_time>::=<time_bounds_spec>
  <reason>::=<text_string>
```

- **Process Quantity Constraints**

```
process_quantity_constraints
  <process_quantity_constraint_id>, <resource_id>,
     <minimum_quantity>, <maximum_quantity>;
  ...
end_process_quantity_constraints;

  <process_quantity_constraint_id>::=<name>
  <resource_id>::=<name>
  <minimum_quantity>::=<digits>
  <maximum_quantity>::=<digits>
```

- **Maximum Rate of Setups**

```
maximum_rate_of_setups
  <maximum_rate_of_setups_id>, <resource_id>, <max_num_setups>,
    <timeperiod>;
  ...
end_maximum_rate_of_setups;

  <maximum_rate_of_setups_id>::=<name>
  <resource_id> ::=<name>
  <max_num_setups>::=<digits>
  <timeperiod>::=<time_spec>
```

- **Non-overlapping Setups**

```
non_overlapping_setups
  <non_overlapping_sutups_id>, <resource_grouping_id>,
     <enforced_time_between_setups>,
     <simultaneous_non_overlapping_sutups_permitted>;
  ...
end_non_overlapping_setups;
```

```
<non_overlapping_sutups_id>    ::= <name>
<resource_grouping_id>         ::= <name>
<enforced_time_between_setups> ::= <time_bounds_spec>
<simultaneous_non_overlapping_sutups_permitted> ::= <digits>
```

- **Environment Variables**

```
environment_variables
   scheduling_horizon      = <scheduling_horizon>  ;
   scheduling_granularity  = <scheduling_granularity> ;
   schedule_rating         = <schedule_rating>  ;
   schedule_strategy       = <schedule_strategy> ;
   time_unit_of_schedule   = <time_unit_of_schedule>  ;
   time_unit_conversions   = <time_unit_conversions> ;
   simultaneous_non_overlapping_sutups_permitted =
     <simultaneous_non_overlapping_sutups_permitted>;
end_environment_variables;
```

```
<scheduling_horizon>    ::= <time_bound_spec>
<scheduling_granularity>::= <time_bound_spec>
<schedule_rating>       ::= <rating_spec>
<schedule_strategy>     ::= <name>
<time_unit_of_schedule> ::= <time_unit>
<time_unit_conversions> ::= <time_unit_conversion_set>
<simultaneous_non_overlapping_sutups_permitted> ::= <digits>
```

References

[Beck 93] H. Beck. TOSCA: A Novel Approach to the Management of Job-Shop Scheduling Constraints. In C.Kooij, editor, *Realising CIM's Industrial Potential.* IOS Press, 1993.

[Berry 91] Pauline M. Berry. *A Predictive Model for Satisfying Conflicting Objectives in Scheduling.* Unpublished PhD thesis, Dept. of Computer Science, Strathclyde University, Glasgow, 1991.

[Chiang *et al* 89] W. Chiang, M. Fox, and P. Ow. Factory Model and Test Data Descriptions: OPIS Experiments. Technical Report Report CMU-RI-TR-90-05, Robotics Institute, Carnegie-Mellon University, 1989.

[Currie & Tate 91] K.W. Currie and A Tate. O-Plan: the Open Planning Architecture. *Artificial Intelligence*, 51(1), 1991.

[Fox & Smith 84] M. Fox and S. Smith. ISIS: A Knowledge-Based System for Factory Scheduling. *Expert Systems*, 1(1), 1984.

[Sadeh 91] N. Sadeh. *Look–ahead techniques for micro–opportunistic job shop scheduling.* Unpublished PhD thesis, School of Computer Science, Carnegie Mellon University, 1991. Available as Techical Report CMU-CS-91-102.

[Smith 87] S. Smith. A constraint-based framework for reactive management of factory schedules. In M. Oliff, editor, *The International Conference on Expert Systems and the Leading Edge in Production Planning and Control*, Charleston, SC, May 1987.

[Smith 89] S. Smith. The OPIS Framework for Modelling Manufacturing Systems. Technical Report Report CMU-RI-TR-89-30, Robotics Institute, Carnegie-Mellon University, 1989.

[Tate 77] A. Tate. Generating Project Networks. In *Proceedings of the fifth IJCAI*, 1977.

[Tate *et al* 93] A. Tate, B. Drabble, and R. Kirby. O-Plan2. In M. Fox, M. & Zweben, editor, *Knowldge Based Scheduling.* Morgan Kaufmann, 1993.