Net-WMS

SPECIFIC TARGETED RESEARCH OR INNOVATION PROJECT

Networked Businesses

D3.1: State-of-the-art of enabling technologies for packing and planning in future WMS

Due date of deliverable: 31 May 2007

Actual submission date: 12 July 2007

Start date of project: 1 September 2006 Duration: 36 months

Organization name of lead contractor for this deliverable: INRIA - partner 2

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)

COVER AND CONTROL PAGE OF DOCUMENT	
Project Acronym:	Net-WMS
Project Full Name:	Towards integrating Virtual Reality and optimization techniques in a new generation of Networked businesses in Warehouse Management Systems under constraints
Document id:	D3.1
Document name:	State-of-the-art of enabling technologies for packing and planning in future WMS
Document type (PU, INT, RE, CO)	PU
Version:	1
Submission date:	12 July 2007
Author: Organization: Email:	François Fages INRIA Francois.Fages@inria.fr

Document type PU = public, INT = internal, RE = restricted, CO = confidential

ABSTRACT :
After a short review of warehouse management systems (WMS), this report describes a state-of-the-art in key enabling technologies for automating packing and planning in future WMS. The report concludes on the main methodology and architecture that will be developed in the Net-WMS project.

KEYWORD LIST :
bin packing, constraint programming, virtual reality, warehouse management systems.

MODIFICATION CONTROL			
Version	Date	Status	Author
0.0	23-04-2007	first draft and call for contributions	F. Fages
0.1	09-05-2007	second draft and call for contributions	F. Fages et al.
0.2	07-07-2007	complete version sent to reviewers	F. Fages et al.
1	12-07-2007	final version	F. Fages et al.

Deliverable manager

- François Fages, INRIA

List of Contributors

- Abder Aggoun, KLS-optim
- Nicolas Beldiceanu, EMN
- Mats Carlsson, SICS
- Filipe Carvalho, WideScope
- François Fages, INRIA
- Philippe Gravez, CEA
- András Kovács, INRIA
- Julien Martin, INRIA

List of Evaluators

- Abder Aggoun, KLS-optim
- Mats Carlsson, SICS

Contents

Chapter 1

Warehouse Management Systems and Networked Businesses

1.1 Supply Chain organization

The term supply chain management was coined by consultant Keith Oliver, of strategy consulting firm Booz Allen Hamilton in 1982. Supply chain management (SCM) is structured into complementary processes of planning, implementing, and controlling the operations of the supply chain with the purpose to satisfy customer requirements as efficiently as possible. The supply chain of a manufacturing enterprise is a world-wide network of suppliers, factories, warehouses, distribution centers and retailers through which raw materials are acquired, transformed and delivered to customers. The organization is so complex that the complete process is structured into a chain of interconnected modules (see Fig. 1.1). Each module can be associated to a business sector. In order to optimize performance of the complete process, supply chain functions must operate in a coordinated manner.

1.2 Warehouse Management system

In order to adequately run a warehouse, there is need of a warehouse management system (WMS) and other software components in order to provide to the end-users an efficient operational systems. In most of cases, a WMS is connected to a host system (ERP, Accounting systems,...). Today, the demand is to share a set of limited information to third parties and especially transport companies.

A WMS is used to control and monitor the main operations of the warehouse like handling reception, processing orders, order picking, order shipping and inventory management. Indeed, the basic logic in handling a warehouse is the capability to exploit a combination of item, location, quantity, inventory, unit of measure, resource, order information and activity to determine where to pick, how to pick and in what sequence to perform these operations. The main functionalities provided in a WMS make possible automating the flow of information and coordinates key activities in a warehouse or in a distribution center, in order to maximize efficiency and increase customer satisfaction.

The process in a WMS is built around the following basic functionalities:

- Inventory Control

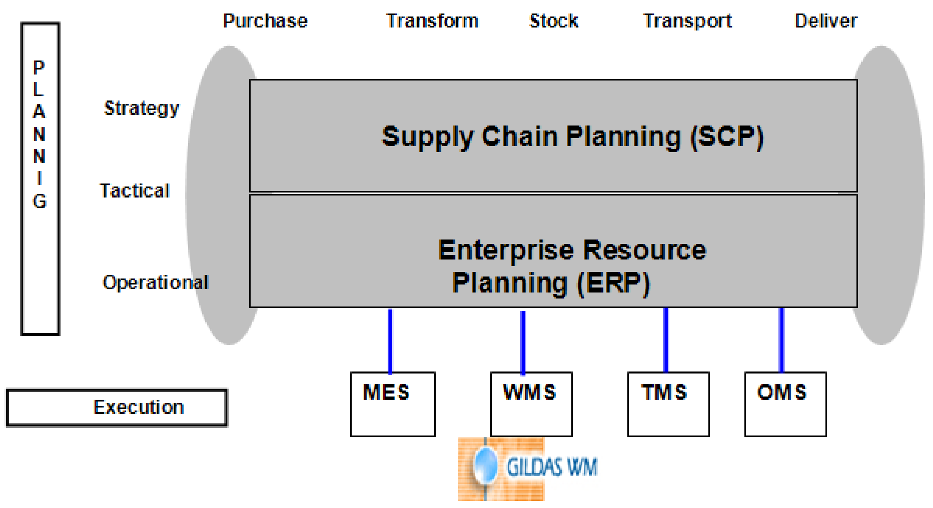


Figure 1.1: Supply chain management organization into modules: Manufacturing Execution System (MES), Warehouse Management System (WMS), Transport Management System (TMS), Order Management System (OMS), Supply Chain Planning (SCP).

- Storage Location Management
- Quality Control Interfacing
- Order Selection
- Automated Inventory Replenishment
- Receiving
- Shipping
- Operator Productivity
- Report Generation
- Preparing activities of orders
- Manual scheduling of activities

Expected new functionalities are :

- Packing activities of orders
- Loading activities
- Vehicle handling (assigning orders to vehicles)
- Gate handling (assigning of vehicles to gates)
- Automatic scheduling of activities

- Intranet capabilities
- Internet capabilities

A WMS application integrates modules in order to provide bar code reading capabilities and even RFID (radio frequency identification) for high tech tracking.

Advantages

A modern WMS will have capabilities to meet the needs of small/medium/large warehouses in several domains from mass storage to specialists in production and distribution, and thus in different business domains : agronomy, hospitals, manufacturing, ...

An efficient deployment of a WMS will bring efficiency and many other advantages among them : reduction of labor costs, paperless, increasing in accuracy, traceability capabilities, reducing of cycle times, reduction in inventory, increasing storage capacity, efficient scheduling of internal / external activities, reduction of lateness and information sharing (a WMS becomes the information system of the warehouse).

Expected new features

A warehouse management system (WMS) is initially a system to control movement and storage of materials within a warehouse. Its role is expanding to include light manufacturing, transportation management, order management, and complete accounting systems. Existing WMS provide advanced features to manage the movement of items within the warehouse, but fail to comply with the increasing demand on *more numerical handling*, such as: how to pack items in a container, how many cartons are needed to pack customer items, how to schedule the manpower to finish the preparation in time, in which order to pack items in a pallet, and the position of pallets in a truck according to the customers to visit. Generally, all existing WMS lack *optimization functionalities* like planning, scheduling, advanced packing tools, optimal filling of containers and trucks subject to delivery constraints.

1.3 Commercially available Warehouse Management Systems

Solution providers in the sector of WMS are very active. The competition is hard. Mainly there are two types of solutions.

Small, standard logistic process

The first category contains solutions for small warehouses where the logistic processes are rather standard. Typical customers are storage warehouses. Stored goods belong to specific customers or to the company. The main activities are :

- Receptions of pallets : reception control, storage in specific zones.
- Command preparation : the customer sends an order to prepare and to deliver to final customer.
- Inventories.

- Generation of reports (daily, weekly or monthly) for accounting.

In some cases, the warehouse looks very large. However, the process remains standard and the number of employees is rather small (1 to 5 persons).

In this category, the contribution of the tools developed in the Net-WMS project will be mainly in the domain of Intranet / Internet. The added value is the capability of customers (the owner of the stock) to access a set of functionalities like :

- Real time current stock.
- Value assessment of the stock.
- Possibility to create online new receptions or new command preparations. Usually such orders are sent by fax.

Complex logistic process

In the second category, the logistic processes are rather complex. The deployment of an application ranges from 3 to 12 months. The following is a list of well-known warehouse management systems:

- **SAP** : The SAP company is world-wide leader in Supply Chain Management Organization. The company provides a complete offer for most components composing a Supply Chain Organization. SAP is well known for its Enterprise Resource Planning (ERP) System. SAP provides also a WMS component which can operate as a stand-alone decentralized system that is independent of a central Enterprise Resource Planning (ERP) System. However, the ERP of SAP is very open allowing WMS providers to connect their software and to focus mainly on the WMS activities.
- **LM7 of a-SIS** : The company a-SIS is an important actor in Europe in the domain of warehouse management systems for the supply chain industry. The company is a subsidiary of SAVOYE (LEGRIS INDUSTRIES group) and is established in France, the UK, Germany, Spain and Italy, with partners the world over. The WMS software of a-SIS is LM7 - a Supply Chain Execution software.
- **Agrostar WMS** : The French company Agrostar is a subsidiary of the group STEF-TFE. The company is one of the leaders in the food sector with the software Agrostar WMS.
- **GILDAS of KLS** : KLS is a highly specialized software company in supply chain and logistics by providing completely integrated Warehouse Management System (WMS) solutions and decision support systems. KLS offers solutions in order to optimize the organization of storage locations, preparation flows of commands, orders, expeditions and information flows within proven software packages GILDAS 2000, GILDAS WE and GILDAS WM.
- **SAGE** : Like SAP, SAGE offers a set of modules in the Supply Chain Organization (ERP, WMS, TMS). Sage Accpac WMS is a competitive warehouse management solution for small and medium-sized warehouses.

- **ORACLE WMS** : Like SAP and SAGE, Oracle offers a set of modules in the Supply Chain Organization (ERP, WMS). Oracle WMS enables warehouse facilities to efficiently and quickly flow goods across a wide variety of business processes- including order fulfillment, manufacturing, service, inbound logistics, counting or replenishment.
- **Manhattan Associates' WMS** : Manhattan proposes a complete solution from Supply Chain Planning to Supply Chain Execution.

The role of a WMS application is evolving due to the market demand. There is a demand to include extra functionalities which are not originally in the scope of a standard WMS. Such functionalities are part of TMS segments. Advanced packing functionalities lack in most existing WMS of the market. Optimization of vehicle loading is part of TMS. Indeed, this functionality is more and more expected in future WMS due to the market demand. Experts try to separate the different businesses (MES, WMS, TMS). However, the business in the WMS is evolving to satisfy the customer demand.

1.4 Networked J2EE architecture

Networked architectures play an important role in current technology information systems, and constitute a crucial aspect of supply chain management. The industry requirements include highly available, secure, portable, reliable and scalable services that integrate distinct enterprise information systems and provide several front-ends for users across a network.

The integration of such services, data and business logic is usually provided by middleware components, which - putting it simply - are server-side software solutions that connect and integrate those systems and resources potentially spread across a network.

As identified in [37], before the advent of J2EE, ”middleware solutions were highly proprietary and restrictive to specific vendors and products, even with limited features and compatibility, and no interoperability or portability across different solutions was provided. Industry standards and common practices were rare and many features were either proprietary or left to the choice of vendors.

J2EE is a middleware architecture for developing and deploying multi-tier, distributed, enterprise scale business applications. Applications that follow the J2EE standards inherently benefit from features such as scalability, portability, reusability, security, and load-balancing.

J2EE represents the maturity and seasoning that middleware technology has undergone over the years by learning from the mistakes of the past and addressing all the essential requirements of the industry. It also provides enough room for future developments. While developing this standard, Sun Microsystems collaborated with other major vendors of middleware, operating systems, and database management systems-including IBM and Oracle.

At its core, J2EE is a set of standards and guidelines that defines how distributed n-tier applications can be built using the Java language. Developers build their applications on the top of these standards while middleware infrastructure vendors ensure compatibility to these guidelines set forth by J2EE . Thus, J2EE applications can be ported and deployed across several application servers, with minimal or no code-level changes.”

As proposed in [55] the platform reduces the cost and complexity of developing these multi-tier services, resulting in services that can be rapidly deployed and easily enhanced as the enterprise responds to competitive pressures.

A J2EE enterprise-level networked architecture addresses several requirements [65], such as:

- Remote method invocations: logic that connects a client and server via a network connection. This includes dispatching method requests, brokering of parameters, and more.
- Load balancing: clients must be directed to the server with the lightest load. If a server is overloaded, a different server should be chosen.
- Transparent fail-over: if a server crashes, or if the network crashes, clients must be rerouted to other servers without interruption of service. This shall happen fast enough according to the business criticality.
- Back-end integration: code needs to be written to persist business data into a database as well as integrate with legacy systems that already exist.
- Transactions: transactions are required to protect data integrity and consistency namely when two clients access the same row of the database.
- Clustering: if the server that contains state crashes, clustering enables such state replication across all servers so that clients can use a different server.
- Dynamic redeployment: software upgrades can be allowed while the system is running. However, this feature may be disabled for performance reasons.
- Clean shutdown: if a server is required to be shut down it must happen smoothly such that clients already posting requests to the server are not abruptly interrupted.
- Logging and auditing: if something goes wrong a log is available for consultation in order to determine the cause of the problem, and help in its debug.
- Threading: having many clients connecting to a server requires the capability of processing multiple requests simultaneously. This means that the server application must be coded to be multi-threaded.
- Object life cycle: the objects that live within the server need to be created or destroyed when client traffic increases or decreases, respectively.
- Resource pooling: if a client is not currently using a server, that server's precious resources can be returned to a pool to be reused when other clients connect. This includes sockets (such as database connections) as well as objects that live within the server.
- Security: the servers and databases need to be shielded from saboteurs. Known users must be allowed to perform only operations that they have rights to perform.
- Caching: many objects are used by the same clients over and over again. There is no reason why they should be instantiated and destroyed on every client request. There are performance increases when such objects are cached and reused in the same context.

The set of J2EE platform components and tools is very wide. Besides the standard components there are a lot of frameworks, design-patterns, tools, application servers, development environments requiring a developer's attention in order to choose the most appropriate approach and complexity for a project.

Some key technologies and tools include JSP, Servlet, EJB, JDBC, Struts, JSF, Spring, iBatis, Hibernate, JAXB, MVC, Jboss, IBM WebSphere [2], Bea Weblogic among many others.

A state-of-the-art architecture would then include several options for its tiers, featuring a Model-View-Controller framework (e.g. Struts, or Java Server Faces), a set of middleware components like Enterprise Java Beans, an Object-Relational Mapper like Hibernate, or any other combination of tools appropriate for the implementation of each project.

Chapter 2

Bin Packing Problems and Algorithms

The bin packing problem is a classical combinatorial optimization problem that has contributed as a study case to the development of the theory of algorithms and complexity since the early 1970's. This NP-hard problem is furthermore of great practical importance. It is the central problem of many industrial applications, such as loading and placement problems for which the bins represent trucks, containers or warehouses, and the items represent the pieces to be loaded according to weight, length, surface or volume constraints. It is thus a core problem for the numerical handling and optimization of warehouse management. It can also be used to model cutting stock problems, where the bins represent the standard length, surface or volume of some material (cable, paper, etc.) in which the items must be cut.

In the first section, we review different bin packing problems according to their dimension:

- ranging from one-dimensional (1D) linear bin packing problems;
- to 2D problems for packing squares, rectangles, or polygons, without or with rotations, either discrete or continuous, into rectangular bins;
- 3D problems for packing objects in space, such as in container loading problems;
- higher-dimensional bin packing problems, including for instance the time dimension for dealing with scheduling constraints in a uniform setting.

In all these cases, one can distinguish off-line problems, where all the bins and the items to pack are known in advance, from on-line problems, where a stream of items has to be packed into a stream of bins.

Then in the following sections, we review different algorithmic approaches for solving bin packing problems, either approximate algorithms, or exact algorithms that include optimality proofs w.r.t. some optimization criterion.

2.1 Bin Packing Problems

2.1.1 One-dimensional Bin Packing

The *one-dimensional bin packing offline problem* (1BP) is, given
Input: N items of length $L_1, \ldots, L_N$ to pack in K bins of length L
Output: a packing if one exists (decision problem), or the minimum value K^* of K (optimization problem).

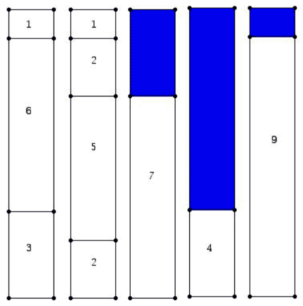


Figure 2.1: Solution to a 1BP problem.

The decision problem is obviously in NP, as by guessing the assignment and position of the items in the bins, one can easily check in polynomial time whether they constitute a solution. The decision problem has been proved NP-complete [36], the optimization problem is thus NP-hard.

The one dimensional bin packing problem is thus already of a high theoretical complexity, and the same in fact as the one of the higher-dimensional problems studied in the following sections. It is worth noting that the one dimensional problem can already express the core problem of truck loading for instance.

The corresponding *online problem*, noted 1BP$^+$, models the situation where the items arrive in some order, and must be packed in a bin as soon as they arrive, without knowledge of the remaining items. The performance of online algorithms is measured by comparing the number of bins used at the end of the sequence of items, to the optimal value of K in the offline problem. Their worst-case (resp. average) *incremental complexity* is measured by the maximum (resp. average) number of operations required to process one item. Their worst-case (resp. average) *amortized complexity* is obtained by dividing the worst-case (resp. average) complexity for processing one sequence of items by the number of items.

2.1.2 2D Bin Packing

In two dimensions, the items and the bins may have square, rectangular, polygonal or irregular shapes. Furthermore, the items to pack may either have a fixed orientation, or can be rotated by either orthogonal rotations or arbitrary rotations. All these problems have the same NP-complete theoretical complexity (under the assumption of a finite number of possible rotation angles in the latter case).

In the common case of rectangular shapes, these different possibilities give rise to the following 2D bin packing problems:

- 2BP for packing oriented rectangular bins into rectangular bins (see Fig. 2.2;
- 2BPo when orthogonal rotations are allowed (Fig. 2.2 if the items $(8, 4)$ and $(4, 8)$ are not distinguished);
- 2BPr when all rotations are allowed (see Fig. 2.3.

A significant portion of research focuses on packing rectangular items, however some applications require considering items of irregular shapes. The industrial requirements, typical policies, and computational approaches to packing or cutting irregular shapes have been surveyed by

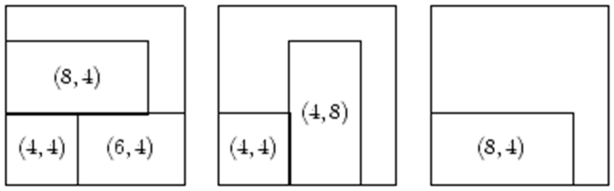


Figure 2.2: Solution to a 2BP problem.

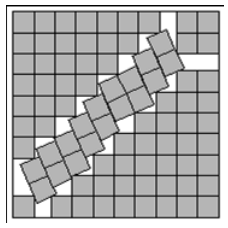


Figure 2.3: Best known solution to the 2BPr problem of packing 86 squares in a square of minimum real-valued size [35].

Dowsland & Dowsland [28]. A possible way of substantially simplifying the geometrical aspect of the packing problem is approximating the shape of items by a union of boxes. Beldiceanu and Carlsson take this approach in their constraint programming model.

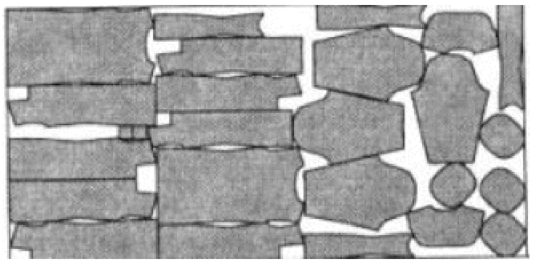


Figure 2.4: A polygon packing from the clothing industry [23].

The 2D bin backing problem is also closely related to the *cutting stock* problem, where a set of rectangular shapes has to be cut from a given rectangular sheet. In practical applications of the cutting stock problem, there are some typical requirements on the cutting patterns within a sheet. The most frequent such feature is perhaps the requirement of *guillotine cuts*, which states that cuts have to be made horizontally or vertically from one border to the opposite one. Furthermore, cutting machines are often constructed to have a certain number of *stages*. Each stage is only able to perform either horizontal or vertical cuts but not both, and pieces having passed a stage may not be put back to a previous stage. These conditions limit the nesting of horizontal and vertical cuts and, thus, the maximum height of the cutting tree of each bin. Fig.2.4 depicts a polygon packing from the clothing industry [23].

2.1.3 3D Bin Packing

In three dimensional space, the 3BP problem consists of packing rectangular boxes into rectangular containers (see Fig. 2.5). This problem is NP-complete. Here again, one can consider the variants 3BPo allowing orthogonal rotations, 3BPr allowing arbitrary rotations, 3BP$+$ for the on-line problem of packing a stream of boxes into a stream of containers, etc.

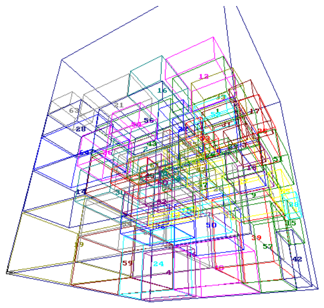


Figure 2.5: Solution to a 3BP problem.

The core container loading and pallet loading problems are 3BP problems, but while in bin packing the only objective is to achieve high volumetric use, the real problem requires additional constraints of gravity, as well as some additional requirements coming from the practical applications. Examples of such additional requirements are the stacking conditions, cargo stability, visibility and accessibility considerations, or different rules that address the prevention of in-transit damage or constrain the loading sequence. The significance of considering these practical requirements in addition to the volumetric use, especially the weight distribution in container loading, has been emphasized by Davis and Bischoff [24]. These requirements depend on the given application, especially on the means of transport. For example, in aircraft loading and when loading a container that will be lifted by a crane, the center of gravity of the cargo has to be located in the center of the container. Since the length of the container (or the aircraft hull) is much greater than its width or height, the longitudinal balance is the most important issue. This motivated various authors to approach this problem from a one-dimensional perspective, see, e.g. [54].

2.1.4 Higher Dimensional Bin Packing

It is worth noticing that the problem of solving a series of bin packing problems over time can be represented as a single bin packing problem with an extra dimension for time. An $(n+1)$D bin packing problem can thus be seen as a planning problem for solving nD packing problems over time.

This approach provides a uniform setting for dealing with some scheduling aspects in practical packing problems, with pure packing techniques. Furthermore, the same distinction between off-line and on-line problems applies to higher-dimensional bin packing problems, to distinguish between the cases where the items to pack at a given time are known in advance or not.

2.1.5 Bin Design Problem

The bin design problem is a variant of the bin packing problem where one wants to determine the best shape of the bins in order to pack a given set of items and minimize the waste of space. Korf [46, 47] considers the problem of optimal rectangle packing, i.e., finding an enclosing rectangle of minimum area that will contain a given set of rectangles. In the case of square items with rotations, note that Fig. 2.3 in fact depicts a solution to the square bin design problem.

Another view of bin design is addressed in the *bin stretching* problem, where the number of bins is given, and the maximum of the bin sizes has to be minimized. Its one-dimensional on-line version has been investigated by Azar & Regev [4] and Epstein [29]. A closely related problem is *extensible bin packing* [21], where unit-size bins are available for unit cost, but irregular bin sizes can be applied as well for a cost proportional to the bin size. Then, it is the total cost of the bins that has to be minimized.

2.2 Heuristics and Greedy Algorithms

Approximation algorithms for one dimensional bin packing offline and online problems are reviewed in details in [20], together with their time complexity in both the worst-case and the average case. These algorithms are based on heuristics for selecting a (one dimensional) bin for each item, assuming an ordering on both the items and the bins. The main selection heuristics are the following:

- *First Fit*: pack each item in order, in the first bin which can contain it;
- *Best Fit*: pack each item in order, in the first bin which will have the least amount of space left after accommodating it;
- *Square Size*: pack each item in the first bin that minimizes the square size of the packing $ss = \sum_{h=1}^{B-1} N(h)^2$, where $N(h)$ is the number of partially filled bins whose contents have total size equal to h and B is the bin capacity;
- *First Fit Decreasing*: same as First Fit with items sorted by decreasing size;
- *Best Fit Decreasing*: same as Best Fit with items sorted by decreasing size.

Note that the first three heuristics can be applied to online problems, while the Decreasing heuristics are obviously restricted to offline problems.

2.2.1 One Dimension

In 1BP problems, the performance of the greedy algorithms based on these heuristics has been analyzed. Let K denote the minimum number of bins required to solve a given 1BP problem of n items.

Theorem 1 *[20] The packings generated by either First Fit or Best Fit heuristics use no more than $\lceil 17/10\, K^* \rceil$ bins, and those generated by either First Fit Decreasing or Best Fit Decreasing heuristics use no more than $11/9\, K^* + 4$ bins,* First Fit and Best Fit heuristics thus provide essentially the same worst-case solutions at 70% of the optimal solution, while the Decreasing heuristics for offline problems improve the performance to 22% of the optimal solution. On the other hand, the square size heuristics ensure a surprising sublinear waste in bounded optimal waste problems.

Theorem 2 *[22] The packings generated by the square size heuristics has $O(log\ n)$ waste on any distribution with bounded optimal waste. There is a variant of the heuristics that has bounded expected waste on bounded optimal waste distribution, and $O(\sqrt{n})$ waste on perfectly packable distributions.*

With appropriate data structures, these algorithms can be implemented with $O(n\ log\ n)$ time complexity [20], and $O(nB)$ for the square size heuristics [22].

2.2.2 Higher Dimensions

In higher-dimensional bin packing problems, the heuristics have to cope not only with the selection of the bin, but also with the placement and orientation of items within the selected bin. In one dimensional bin packing it is indeed trivial to check whether a set of items fit into a bin by comparing the total size of the items to the capacity of the bin. However, when packing higher dimensional items with rigid shapes, this subproblem becomes NP-complete, and to solve it, items have to be effectively placed in the bins. Most of the frequently applied search strategies belong to the class of *absolute position* or *relative position* heuristics.

Absolute position strategies bind the position of one item inside the bin in each phase of the search. A typical way of reducing the number of possible positions to consider is using the *leftmost-downward* strategy: in each step, an item is placed with its left side touching either the right side of the previously placed item or the side of the bin, and its bottom side touching one of the previously placed items or the bottom of the bin (see Fig. 2.6). A drawback of the absolute position heuristics is that the huge number of possible placements might not be reduced in such a way when additional constraints are present in the model.

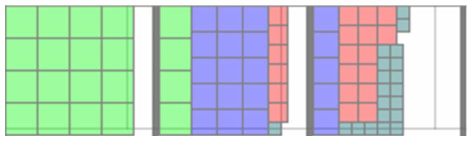


Figure 2.6: A solution obtained to a 2BP problem by the First Fit Decreasing / Left Most Downward heuristics.

Relative position heuristic decide in each search step the relative position of a pair of items in the solution. The number of possible relative position relations for k-dimensional rectangle packing is $2k$, e.g., in two dimensions it is *left*, *right*, *above*, or *below*. In most bin packing problems, the complete specification of the relative positions can easily be converted into a solution with bound absolute positions. While relative position heuristics are more robust with respect to side constraints in the model, difficulties arise when trying to define the relative positions of items with complex shapes.

The performances of heuristics for 2D bin-packing are compared in [50], while [17, 16] describe some approaches for solving the 2D bin packing problem without explicitly solving the placement problem.

For container loading problems, Davis and Bischoff [24] propose a loading heuristic that achieves high space use and correct weight distribution, in order to find solutions such that the center of gravity is located in the center of the loaded container, thereby ensuring an even weight distribution of the cargo.

For packing problems with irregular shapes, while automated solution methods are often based on trying to capture human experts' knowledge, various heuristics have also been proposed to improve the search for solutions. Recently, Burke et al. [12] have proposed a novel heuristic for packing items of irregular shape in two dimensions. The heuristic is based on the notion of no-fit polygons, which define the positions in which a pair of polygons touch without overlapping [13]. The method applies to convex and concave shapes bordered by straight lines and arcs, potentially with a hole in them. Combined with a tabu search, the above heuristic is considered to be the most efficient solution method known for irregular packing.

2.3 Constraint Programming

In the context of finite domain constraints, Beldiceanu and Contejean [BeldiceanuContejean94] introduced the primitive constraint `diffn`, which states that a set of n-dimensional parallelepipeds are mutually non-overlapping. Aggoun and Beldiceanu [AggounBeldiceanu93] introduced the global constraints `cumulative`, and showed that one of its important uses is to provide necessary conditions for placement problems. Beldiceanu and Carlsson [BC01] introduce a generic pruning technique called *sweep*, and show how it can be used to propagate to the `disjoint2` constraint, which is the 2D case of `diffn`.

Shaw [71] introduced a novel constraint for one dimensional bin packing problems, which incorporates several propagation rules. The rules infer about into which bin certain items can be packed based on the solutions of subset sum problems. The proposed CP-based approach, when used in combination with dominance rules and a standard search strategy, is competitive with dedicated exact algorithms for the 1D bin packing problem.

A series of papers from Carlier et al. and Clautiaux et al. [15, 17, 16, 19] address different variants of the two dimensional bin packing problem. [15] introduces reduction procedures and lower bounds for the 2BPo problem. Reduction rules are based on so-called *identically-feasible functions* (IFFs), i.e., transformations of the problem instance that do not change the objective value. The proposed IFFs decrease problem size, e.g., by removing some small items and increasing the size of larger items in turn. In addition, new lower bounds based on *dual feasible functions* (DFFs) are presented. In [19], a strong lower bound based on DFFs is proposed for the variant where the orthogonal rotation of rectangle-shaped items is permitted and the bins are squares. The lower bound can be generalized to higher dimensions and rectangular bins as well, but this results in a loss of tightness. In [17], the same authors present how the above techniques can be exploited in solving the two dimensional orthogonal packing problem (2OPP), which consists of determining whether a set of rectangular items can be packed into a larger rectangle. [16] introduces an exact method for the 2BPo problem, which is based on the iterative decomposition of the bin packing problem into several instances of 2OPP. The latter are then solved by the algorithm proposed in [17].

Claudiaux et al. [18] propose a constraint programming approach to 2OPP. The proposed CP model exploits the cumulative scheduling relaxation of the packing problem, and makes use constraint propagation and search techniques familiar from cumulative scheduling to solve them efficiently. A generalization of energetic reasoning to two-dimensional orthogonal packing problems, and a new pruning technique based on subset sum problems are presented. The CP-based approach improves on the results achieved previously by the same authors in [17].

For pallet loading problems, constraint solvers are used with a tree search procedure for positioning a new box at each step. The choices of the box and of its place can be made according to different heuristics. One search strategy is the so called G4 heuristics [69] which recursively divide the placement space into four huge rectangles possibly with a small waste in the center. Problems up to 50 boxes can systematically be solved exactly in a reasonable time limit. Beside the use of an appropriate heuristics, the key point is the use of upper bounds on the maximum number of boxes that can be packed. Some bounds like the Barnes [8] and the Keber [44] bounds consider the geometric structure of the problem. Some other bounds are obtained by solving a linear programming problem [42]. Komarnicki and Lahrichi [KomarnickiLahrichi83] report on the *black hole* heuristic for the 2D strip packing problem. Chazelle [Chazelle83] reports on an efficient implementation of the *bottom-left* heuristic for the rectangle packing problem. Lim et al. [LimRodriguesYang05] report on heuristics for 3D container packing. Lesh et al. [LeshMarksMcMahonMitzenmacher04] report on an effective pruning method for rectangle packing problems with zero slack. Aggoun and Beldiceanu [AggounBeldiceanu93] report on an effective search strategy for placement problems:

- For each dimension d, for each object o, fix the origin of object o in dimension d.

Furthermore, if the placement space has zero slack, i.e. no hole is allowed, this strategy can be replaced by:

- For each dimension d, for each point x, find an object to cover point x in dimension d.

2.4 Mixed Integer Linear Programming

Mixed integer linear programming (MILP) approaches to the 2D bin packing (or cutting stock) problem have been proposed in [60]. Fasano [34] proposed a MILP approach to solving the 3D single bin packing problem with orthogonal rotation allowed and the center of gravity location constrained in all three dimensions. Scheithauer & Tarno [68] describe a MILP model for packing convex polyhedra in two dimensions, with possible generalization to higher dimensions.

Pisinger and Sigurd [58] proposed a hybrid MILP-CP approach based on Dantzig-Wolfe decomposition for solving the 2D bin packing problem. Their master problem addresses the assignment of items to bins by using MILP techniques. Subproblems deal with the placement of the items in a single bin, and are solved by constraint programming. When CP proves the infeasibility of a subproblem, then valid inequalities are fed back to the master problem. The single-bin packing subproblems are solved by techniques that have been introduced originally in [57] for solving 3D packing problems. In this representation, beyond the classical x_i and y_i variables that stand for coordinates of the items in the bin, there is a variable $M_{i,j}$ assigned to each pair of items to denote the relative position of the two rectangles, with an initial domain $\{$*left, right, above, below*$\}$. A simple propagation rule is proposed to tighten the domains of the variables $M_{i,j}$.

2.5 Local Search Methods

Local search methods are often key techniques for handling large size combinatorial optimization problems. These methods make a configuration evolve towards a solution by iteratively performing local modifications. These modifications aim at performing two complementary, yet contradictory, processes:

- intensification, to improve the cost function (combining the degree of satisfaction of constraints with the optimization criterion);
- diversification, to escape from local minima (by degrading the cost function).

A wide variety of local search methods have been developed, differing by the ways, called metaheuristics, these two processes are defined. One can cite:

- Tabu search, where diversification is realized by ignoring for some time the variables which reached a local optimum,
- Simulated annealing, where diversification is implemented by random moves, and controlled by a decreasing temperature,
- Ant colony, where intensification is realized by the memorization of past explorations,
- Genetic algorithms, which are multipoint methods consisting in maintaining a population of configurations with operations for cross evolution.

Concerning container loading problems, Wodziak and Fadel [73] describe a genetic algorithm to minimize the distance of the center of gravity of the cargo from the desired location in one, two, and two and a half dimensions. For pallet loading problems, Alvarez et al. [1] describe metaheuristics based on genetic algorithms and tabu search. Approximate algorithms are based on constructive methods that divide the pallet into blocks or in any recursive way. Note that the problem is usually first normalized in order to reduce the set of possible solutions [26, 27].

It is worth noticing that local search methods can also be combined with other methods, for instance for improving the quality of solutions found by constraint solving or linear programming methods.

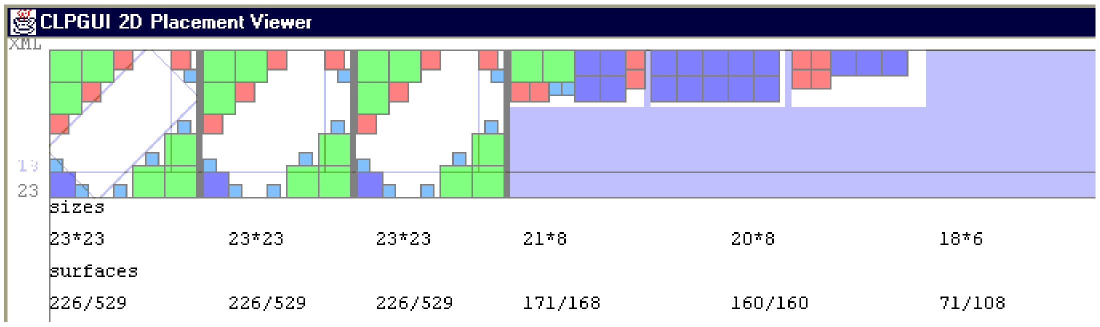
2.6 Optimization and User Interaction

In complex systems, the purpose of an optimization tool is not to take decisions automatically but to provide solutions to the user for helping him to take good decisions. The interaction capabilities with the optimization component are thus crucial, for making it possible to modify some parts of the computed solutions, some constraints of the problem, or some optimization criteria, and use the solver in an incremental fashion.

Fages, Fowler and Sola [31, 32] defined a general scheme for solving incrementally a series of constraint satisfaction goals defined by a succession of additions or deletions of constraints in the problem. A minimal perturbation problem was described formally by El Sakkout, Richards, and Wallace in [67] as a 5-tuple $(Q, a, C_{add}, C_{del}, d)$, where Q is a CSP, a is a solution to Q, C_{add} and C_{del} are constraint removal and addition sets, and d is a function that measures the distance between two complete assignments. Their solving method combining linear and constraint programming is looking for a complete assignment b that minimizes $d(a, b)$ and that is a solution of the new problem arising from Q by adding the constraints from C_{add} and deleting the constraints from C_{del}.

Bartak, Muller and Rudova [9] considered a variant of the minimal perturbation problem in the context of constraint satisfaction motivated by a real-life application of university course timetabling. Here, a minimal perturbation problem (MPP) is a triple $P = (Q, a, d)$, where: Q is a CSP, a is a possibly partial assignment for Q called an initial assignment, d is a distance function defining a distance between any two assignments. A solution to the minimal perturbation problem $P = (Q, a, d)$ is a solution b for Q such that $d(a, b)$ is minimal. The task is to find the largest possible assignment of variables for the problem Q in such a way that it differs minimally from the initial assignment. This new framework supports over-constrained problems as well as hard-to-solve problems. Moreover, this new formulation allows looser changes of the problem formulation like addition and retraction of constraints and variables as well as changes in the variables' domains.

In [33], Fages, Soliman and Coolen describe a generic graphical interface for constraint programs, called CLPGUI, which allows not only visualizing the execution of the program (domains of variables, search tree, solutions, ...), but also interacting with the solver at different levels: variable instanciation, search tree state recomputation, modifications of solutions, etc. It has been used in a 2BPr problem for finding packings with rotation similar to the one depicted in Fig. 2.3, i.e. in a problem where the search cannot be fully automated. The idea was to use the solver in an interactive manner to define graphically, and solve automatically, 2BP problems without rotations but with particular user-defined shapes (see Fig. 2.7), and then apply a squeezing algorithm for computing the minimum size of the bins after insertion (Fig. 2.8).



```
sizes
23*23     23*23     23*23     21*8      20*8      18*6
surfaces
226/529   226/529   226/529   171/168   160/160   71/108
```

Figure 2.7: Combined optimization and interaction for solving a 2BPr problem [33].

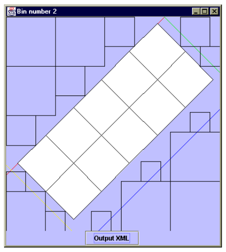


Figure 2.8: Item insertion with rotation and automatic squeezing of the bin.

CLPGUI has also been used experimentally for the automatic placement of furnitures in an office (see Fig. 2.9), with the capability of modifying a solution manually (Fig. 2.10), and computing a new solution that minimizes the distance to the user-modified configuration (Fig. 2.11 and 2.12).

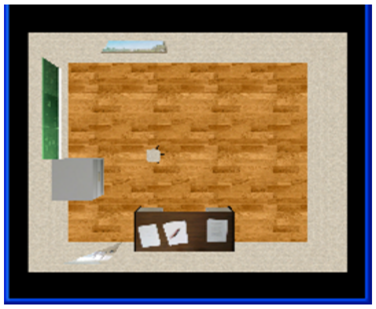


Figure 2.9: Automatic placement of furnitures in an office [33].

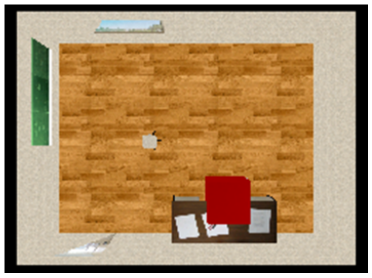


Figure 2.10: User modified placement.

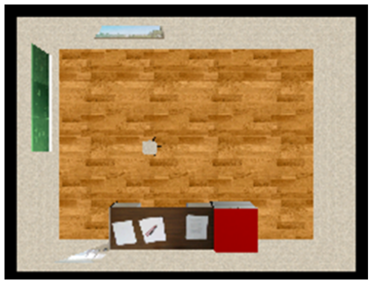


Figure 2.11: Automatic computation of the closest solution.

As an alternative to using an exact method for finding a solution that minimizes the distance to the user-modified configuration, approximate methods can also be considered for this task, such as for instance local search methods starting from the user-modified configuration, or computing a new solution using the user-modified configuration as a heuristics.

2.7 Existing Tools

There are a few tools available on the market dedicated to palletizing and truck loading applications. One can mention:

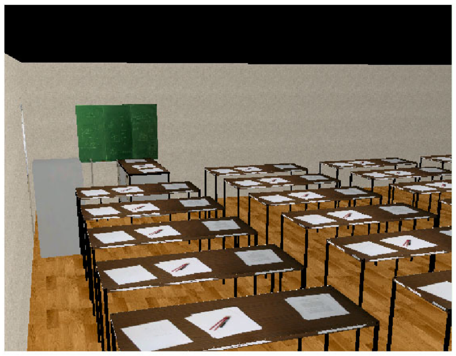


Figure 2.12: Combined optimization and interaction for placing furnitures in an office [33].

- **TOPS:** Engineering Corporation offers packaging software and truck loading software for packaging and distribution professionals. TOPS Pro palletization or pallet software provides package design, pallet layout and compression analysis tools to build mixed pallets and optimized pallet patterns. The MaxLoad Pro container loading and cargo load planning software outputs efficient, 3D load plans for trucks, containers, trailers, railcars to optimize freight layout and cube utilization.
- **CAPE:** CAPE Systems offers a comprehensive range of software products and support services designed to optimize the flow of products through the supply chain. Their Palletization & Packaging Design Software CAPE PACK features pallet patterns building, new case size creation, new product packaging design, strength of corrugated board testing, and in-store displays.

Chapter 3

Knowledge Representation Languages for Packing Problems

To be widely used, and easily maintainable, the optimization components in a WMS should rely on a declarative knowledge representation language for stating the constraints of the problem and the business expertise. One objective of the Net-WMS project is to develop a markup language, called PackML, as an exchange format for defining packing problems. In this chapter, we review some existing languages related to placement problems.

3.1 The Rectangle Placement Language (RPL)

A declarative language intended to specify Very Large Scale Integration (VLSI) layouts was introduced in [64]. More precisely, RPL allows to express placement constraints between

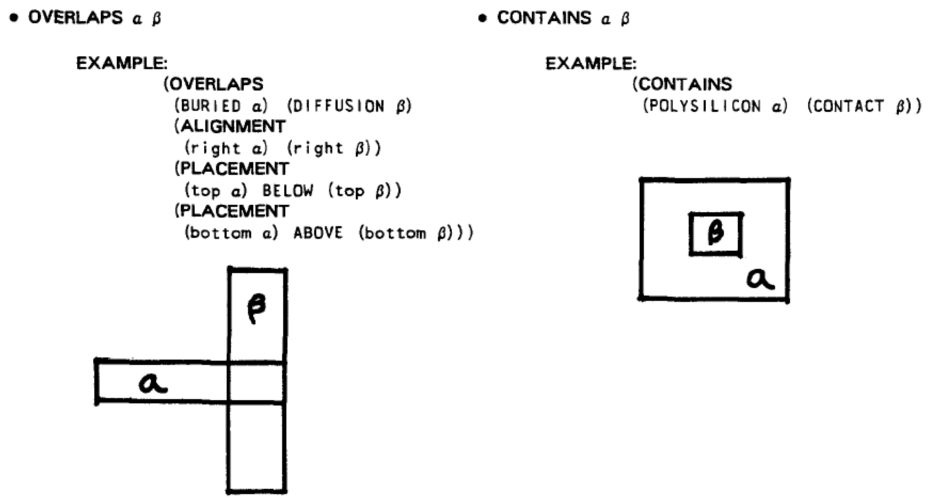


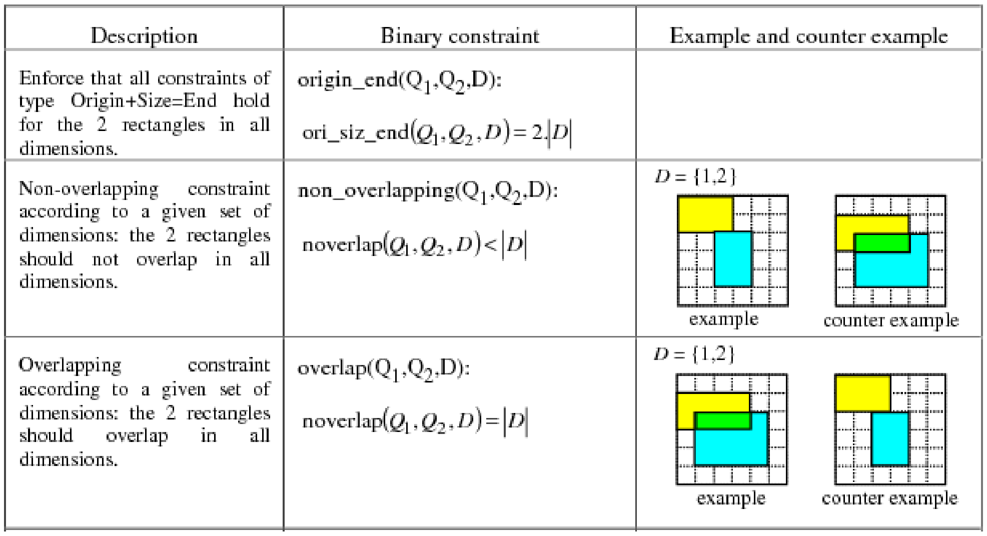
Figure 3.1: Two constraints of the Rectangle Placement Language.

rectangles, under the VLSI layout context, which are solved by specific compaction algorithms. The RPL language is composed of 11 relations which are either VLSI specific like `INSTANTIATE submodule prototype context` or general like `CONTAINS a b` or

`OVERLAPS a b` as represented in Fig. 3.1. RPL interacts with VEXED, a rule-based expert system whose goal is to provide advice and information to a designer during the construction of a VLSI design.

3.2 Elementary Constraints between Multi-dimensional Rectangles

[10] presents a set of binary constraints between multi-dimensional rectangles in a discrete space like for example the followings ones (2D): `non_overlapping(Q1,Q2,D)` (see Fig. 3.2), `include(Q1,Q2,d)`, `distance(Q1,Q2,D)`, `before(Q1,Q2,D)`, `touch(Q1,Q2)`, `on_top_of(Q1,Q2,d)` or `gravity(Q1,Q2,d)`, where `Q1` and `Q2` denote rectangles, `d` a particular dimension and `D` a set of dimensions.



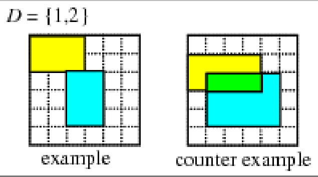
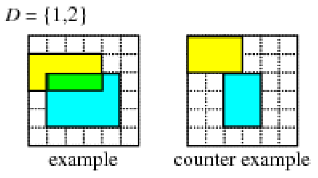
Description	Binary constraint	Example and counter example
Enforce that all constraints of type Origin+Size=End hold for the 2 rectangles in all dimensions.	origin_end(Q_1,Q_2,D): $\text{ori\_siz\_end}(Q_1,Q_2,D)=2.\lvert D\rvert$	
Non-overlapping constraint according to a given set of dimensions: the 2 rectangles should not overlap in all dimensions.	non_overlapping(Q_1,Q_2,D): $\text{noverlap}(Q_1,Q_2,D)<\lvert D\rvert$	$D=\{1,2\}$  example / counter example
Overlapping constraint according to a given set of dimensions: the 2 rectangles should overlap in all dimensions.	overlap(Q_1,Q_2,D): $\text{noverlap}(Q_1,Q_2,D)=\lvert D\rvert$	$D=\{1,2\}$  example / counter example

Figure 3.2: Some binary constraints between 2-dimensional rectangles.

3.3 The Region Connection Calculus (RCC-8)

RCC-8 [61] is a general framework for topological knowledge representation and reasoning. In RCC-8, one distinguishes the eight following topological relations between two regions (see Fig. 3.3): `disjoint`, `meet`, `overlap`, `equal`, `covers`, `coveredby`, `contains`, `inside`. It should be noted that these relations are jointly exhaustive and pairwise disjoint: any two spatial regions are in one and exactly one of the RCC-8 relations.

3.4 Business Rules

In many industry fields, condition-action rules are considered as a natural knowledge representation language because they allow to express clearly, flexibly and concisely the expertise related to the processes of an activity or a business. At the time of expert systems vogue, several rule-based systems were developed in the packing domain. For instance, LeMaster [48] presented an expert system for loading semi-trailer trucks. The system applied rules extracted from the knowledge of

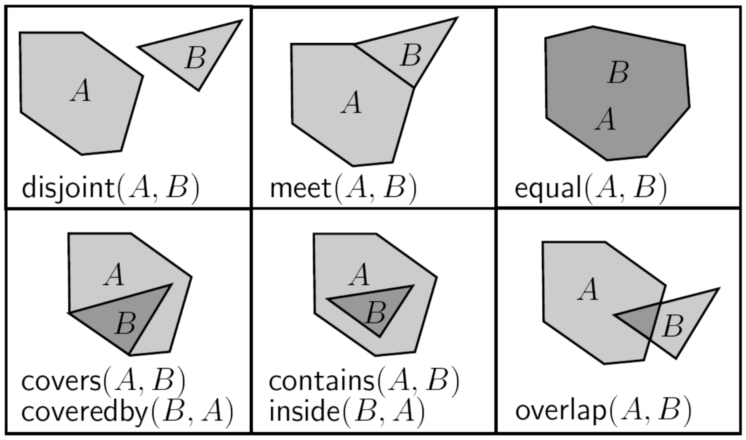


Figure 3.3: The eight relations of the RCC-8 calculus [61].

human experts to prepare loading plans, according to two main criteria: the volumetric efficiency and minimising the risk of in-transit damage to the load.

Nowadays, business rules are considered independently of their possible execution in a precise system [40]. It should be noted that such rules express two kinds of knowledge: a declarative knowledge that gives information about what a solution must satisfy, and a procedural knowledge that provides indications about how to build a solution. Although business rules do not have a formal definition, they are a popular means of expressing knowledge in industry: they are written in natural language, and they can be validated one by one by business people. A group of the W3C is currently working on their standardization in an XML exchange format.

For container loading or pallet loading problems, business rules constitute a natural knowledge representation language in industry. They are for instance used by PSA to formalize its business expertise about container loading, with rules such as :

”All boxes must be stacked in decreasing order of weight”
If P_1 is on top of P_2 **Then** P_2 is heavier than P_1

or

”Over the container length, the weight supported by one half must not exceed 10% more than the weight supported by the other half”
If S_1 is the set of boxes in the left half-length of C,
and S_2 is the set of boxes in the right half-length of C
Then the maximum weight of S_1 and S_2 is less or equal to
1.1 times the minimum weight of S_1 and S_2.

The general problem of translating business rules into constraint programs has not been much studied up to now however. This problem is recognized as an important issue to widen the scope of constraint programming in industry, where the state-of-the-art ranges from programming from scratch to the filling of templates for some predefined problem patterns. It is one clear objective of Net-WMS to make advances on this topic in the framework of packing problems, with the definition of the PackML markup language, and of its computational interpretation as a constraint program.

Chapter 4

Visualization and Interaction in Virtual Reality

4.1 Virtual reality and its potential benefits

From a general point of view, Virtual Reality (VR) may be defined by three keywords:

- Virtual environment
- Interaction
- Immersion.

A virtual environment is a computer-driven model describing objects (manufactured products, packages, trucks ...), human beings (or animals), complex systems (like a warehouse, a factory) or collections of data (for instance, geographical data). It may duplicate an existing environment or embody realistic components that do not presently exist (typically, a future product). It may also partly or completely result from imagination. As an essential point, it has to simulate the behavior of its digital components according to some well-defined realistic laws (like those of optics or physics) and/or artificial rules (like packing rules), under various external conditions (lighting, forces ...). The difference between virtual reality and computer simulation or computer graphics mainly rests in the interactive capability of the virtual environments that must react to external events (typically user inputs) without perceivable delays (interactive simulation).

Actually, the core feature of virtual reality is to allow one or several humans (the human players) to interact with a virtual environment:

- By controlling some of the components or parameters of the virtual environment (the point of view on the virtual scene, the position of an object or the behavior of a virtual human)
- By perceiving the virtual environment through a number of human input channels (vision, sound, touch ...).

Virtual reality looks for making the best use of natural human input channels and interaction modalities (like stereoscopic vision, computing images with respect to the human player's point of view or controlling virtual objects with hand or body motions). It thus more or less creates the feeling of being immersed inside the virtual environment. The interactions of the human players with the virtual environments thus become highly intuitive.

Conversely immersion hides the real surrounding of the humans interacting with the virtual environments. Augmented Reality (AR) techniques have thus been developed to combine the perception of both the real world and virtual components with which the human player may interact.

By providing the human players with combined action and perception means, VR allows an intuitive understanding of the modelled virtual system. It is always easier to master a phenomenon when you can play with it at will. Implementing feedbacks that are impossible in the real world increases this advantage. For example, in a virtual environment you can ”see” through the walls of a container. The first potential benefit of VR thus rests in supporting communication. Nowadays, interactive digital mock-ups are universally recognized as a tremendous tool for intuitively describing a product or an assembly (as a package) to non-specialized people. In the industrial world, they are currently used:

- within a company to boost collaboration between peoples of different trades (mechanical engineers, human factor experts, marketing staff, managers ...)
- with customers to publicize the main features of a new product
- with business partners (subcontractors ...) or public authorities (dealing with policy, safety, ethics ...). Since a digital mock-up is only located in computer memories, it can be also utilized for work sessions or project reviews involving persons that are geographically remote (shared digital mock-ups).

Besides communication, digital mock-ups fulfill the same purpose as physical mock-ups: to validate a product or an assembly at an early stage in the design or planning process. They are simply easier to handle and may be subjected to a larger set of tests and experiments. Through VR and AR techniques, future users may be ”projected” near (or inside) the digital mock-up or the digital mock-up may be ”projected” into the real world, in order to assess the efficiency of the designed object. Using interactive simulation, its main functions may be experimented in standard or extreme operating conditions. In theory, it is thus possible to test:

- the validity of a packing plan
- the associated packing and unpacking procedures
- the package behavior during transport
- ...

Digital mock-ups are also relevant to studying & designing complex systems like factories (the digital factory concept is currently gaining in importance) or urban environments. They are similarly valuable to define, compare and optimize numerous scenarios in such complex domains as public safety (accident prevention, fire fighting ...) or security (facing a terrorist attack).

The above-mentioned applications of virtual reality technology are centered on digital mock-ups. Digital mock-ups formally existed before VR (in computer simulation and computer graphics), but were only used by experts. The revolutionary impact of VR stems from the fact that non-specialists may easily handle interactive digital mock-ups. It thus opens the path for exceedingly interactive and reactive design / planning processes involving all the persons associated with the final result.

Apart from interactive digital mock-ups, a perhaps more innovative use of virtual reality (also more difficult to manage in the professional world) consists in relying on technology to build ”virtual experience”: making persons familiar with existing, inaccessible or conceptual worlds. This may clearly have a tremendous impact on training (up to teaching manual skills with haptic devices), assisting staff (through ”AR user-manual” directly interacting with the real world) and collaborative working (through remote virtual presence). The industry is already considering VR-based training tools to train the workers and the maintenance staff.

Finally, let us stress that virtual reality do not necessarily require expensive and complex hardware equipments implementing sophisticated man-machine interactions. An up-to-date computer with good graphical performances linked to a low-cost 6-dimensional input device (or even a standard mouse) is sufficient for most applications. This constitutes the basic configuration considered in this project and the following sections focus on the domains of virtual reality that are directly relevant to Net-WMS.

4.2 Physical objects in virtual reality

As mentioned above, virtual reality implies simulating the physical behavior of digital objects without any delays perceivable by humans. The term interactive simulation designs the algorithms and techniques utilized for this purpose. A number of interactive simulation software libraries are now available, either commercially or as open source code. All these tools perform up-to-date rigid-body simulation and provide collision detection, contact generation and dynamic response. The main commercial libraries are Vortex from CM-Labs, Novodex from Novodex, Havok from Telekinesys and VPS-PBM from Boeing. The main open source engine is ODE (Open Dynamic Engine). These software components provide robust, ready-to-use codes for rigid-body dynamics and collision detection.

4.2.1 Collision detection and contact generation

All these libraries perform real-time collision detection in a hierarchical way starting with crude, but fast, detection of non-collisions, followed by slower, but accurate, detection of collisions between objects that are close. For example, Vortex features a proprietary implementation of a far field processor that accepts hints as to the distribution of objects in space and automatically divides objects into sets upon which collision detection must be performed. To detect collision overlap, it relies on a number of collision types: primitives (sphere, box, cylinder ...), convex mesh, mesh, triangle list and others. As a general rule, collisions between primitives constitute the fastest collision type and involving smooth surfaces, they generate very stable contacts. Users are thus advised to use primitives wherever possible in simulations to obtain these benefits.

In Vortex and ODE, real-time physical simulation features a fast and optimized convex mesh solution that implements a GJK (Gilbert-Johnson-Keerthi) algorithm [14] [38] [56] [45]. The essential idea is to exploit the coherence of spatial relations between convex objects to allow for fast elimination of non-intersecting objects. Utility functions are provided to create a convex collision object from a mesh by building its convex hull.

The mesh-mesh algorithm implemented in real time physical simulation is proprietary in all commercial libraries. Creating good contacts between two meshes is very complex and it can be difficult to determine the difference between sharp edges actually present in an object (the corner of a box, for example) and the piece-wise linear approximations to smooth surfaces typical of meshes. In the first case, there is a natural and required discontinuity of contact, while in the second contacts must be smoothed to avoid jitter in a simulation. Research teams are currently working on improving this implementation by using topological information to determine the best choice of contacts.

For the mesh case, the approach favored by Havok is based on continuous collision detection. A common problem in many applications that include collision detection is that of temporal aliasing. If objects are moving too fast between collision detection calls, many techniques fail to report a collision. Continuous methods proposed by Stéphane Redon [72] [62] offer a solution to this problem. In addition to being more robust, they have the ability to provide very accurate contact information, which is essential to many simulation applications.

VPS relies on voxels, both as geometric modelling elements and as bounding volumes for collision detection. Since voxel-based collision detection is volumetric in nature, it can detect when an object is wholly embedded in another object. VPS has been carefully optimized for both speed and memory efficiency. PBM refers to the real-time motion simulation part including time critical applications such as haptics (force feedback). These are key capabilities in VPS, because VPS was originally designed for 6-DOF (Degree Of Freedom) haptic rendering. PBM differs from VPS collision and proximity detection. Both involve detecting the collisions of virtual objects, but only PBM generates a collision response and calculates subsequent motion. However, PBM does not return highly detailed collision information, such as which parts or triangles were involved in the collision, because that would consume too much time. Such detailed information is left to VPS collision and proximity detection. For the same reason, PBM is limited to voxel accuracy, and exact-surface accuracy is left to VPS collision and proximity detection.

For dealing with mesh-mesh collisions, Novodex relies on a voxel-based approach similar to the one proposed by VPS.

In academic research, a current trend is to use GPU, see [39] for a recent survey.

4.2.2 Dynamic Response

All these libraries implement a consistent framework for constraints, both kinematic and dynamic. These may be divided into contact and joint constraints, which are handled in a consistent way. Basic Constraint Library is used to take a list of joints defined by the user, contacts generated at run-time and user-defined parameters, and to create a matrix representing the differential equations for the system to be solved.

In all these libraries, except PBM, the core solver uses an LCP (Linear Complementarity Problem) solution. This type of solution for dynamics problems was introduced in Dave Baraff's papers [6] [7]. However, his formulation wasn't completely consistent, and each software tool has created a unique formulation of the friction and contact engine. For example in Vortex and ODE ideas from Uri Ascher and Dave Trinkle have been used to improve methods of returning to a correct solution from an approximation [51] [52] [3]. PBM uses a simple approach which is based on penalty.

4.2.3 Integration with 3rd Party Applications

All these libraries are supplied with utilities to allow easy integration with 3rd party applications. This is done by exchanging the positions and orientations of objects between the two APIs and synchronizing the current time step. Additionally, many utilities are provided to create collision objects from specific graphics objects. ”Bridges” are provided between real-time physical simulation and many scene graphs such as Performer and Virtools. Also, objects in .flt or .obj format may be loaded from disk and used to build a collision object.

VPS-PBM was originally designed for 6-DOF haptic rendering with the Phantom from Sensable. Haption & CEA/LIST have also developed 6-DOF haptic plug-in for Vortex, ODE, Havok and VPS-PBM.

4.2.4 Current research

On the research side (some interesting information may be found in [49]), interactive simulation constitutes a ”hot” topic and numerous software codes have been developed that focus on:

- Collision management in real time on models derived from CAD files (Catia, SolidWorks, ProEngineer ...).

  A lot of commercially available software tools addressing collision detection come from the video game community and are difficult to adapt to interactive simulations with complex industrial models.

- Simulation of the multibody system dynamics.

  This is mandatory to perform simulations involving virtual robotic arms or virtual humans.

- Control of virtual objects/robots inside the interactive simulation, either with passive peripherals (such as a SpaceMouse) or with active ones (as haptic devices).

The management of physical contact models when manipulating virtual objects allow the operator to feel what effectively occurs when operating on real equipments, either for manual or remote controlled tasks.

In this context, research is currently looking into the following orientations:

- Manage the complexity of the CAD models for collision detection.

  This point is essential for handling CAD models and useful work may be done in developing algorithms that are suited to different model formats like point clouds, polyhedrons or Nurbs.

  - For large models, even if the objects are considered as rigid, research has to develop new parallelized detection collision algorithms in order to obtain scalable solutions.
  - To define exact contact models research has to develop new collision detection algorithms which are able to calculate distance between objects and not only penetration depth. Penetration depth is widely used in the game industry.

- Simulate deformable models.

  Managing rigid objects is clearly insufficient to answer the basic industrial requirements. Research has to tackle with deformable objects (like cables, plastic parts and resilient shells) and develop:

  - Dedicated collision detection algorithms for concave deformable models and parallelization of detection collision algorithms. Again to define exact contact models we need distance between objects and not only penetration depth which is widely used in the game industry.

  - Dedicated contact models which can manage friction and the mixing of deformable and rigid models.
  - Dedicated FEM (Finite element code) and associated solver to manage the real time constraint.
  - Real time parallelization of FEM algorithms will be a necessary step to deal with industrial applications.

- Simulate complex physical phenomena.

  Beside classical mechanics, there also exists a strong interest for other phenomena related with thermal diffusion or electro-magnetic shielding.

4.2.5 CEA-List developments

CEA-List is actively engaged in this field. It has in particular developed two software codes focusing on applying interactive simulation techniques to the industry requirements. The first one is a fast collision detection module to efficiently compute Local Minimum Distances (LMD) between CAD objects. Called LMD++ and operating on meshed objects, its main features are a good capability for processing the concave parts that are typically found in CAD model (similar software provided by the game industry are unsuited from this respect) and accurate contact simulation (compared with VPS which is faster but less exact).

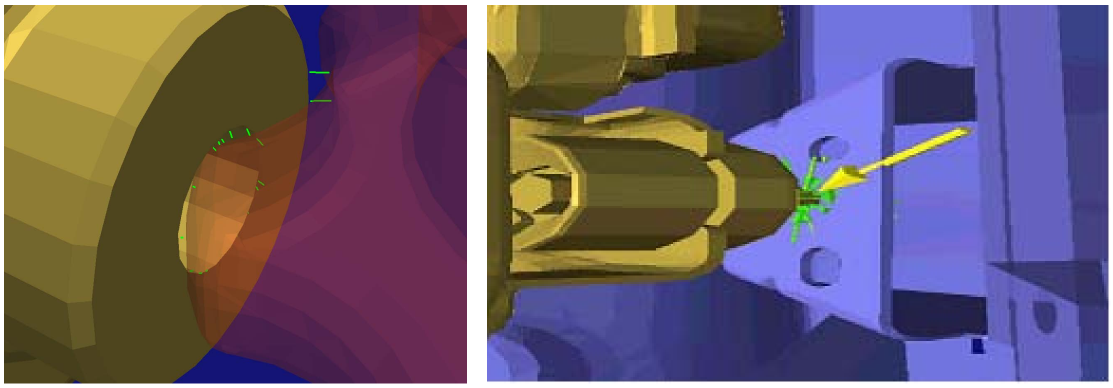


Figure 4.1: (a and b): Collision detection with LMD++.

The second one, GVM (for Generalized Virtual Mechanisms), manages unilateral and bilateral constraints like those experienced when articulated objects interact - GVM is actually a solver for the integration of motion laws.

These two pieces of software are currently accessed through a common API entitled XDE. Current work is progressing on deformable objects (XDE-soft), sound generation (XDE-sound) and virtual humans (XDE-manikin).

4.3 Human workers in virtual reality

Virtual humans are a major task in virtual prototyping approaches. Until recently, all the tests achieved on digital mock-ups were purely technical (strength and thermic analysis, vibrations ...) and the human aspects had to wait for the physical models to be available (which can be

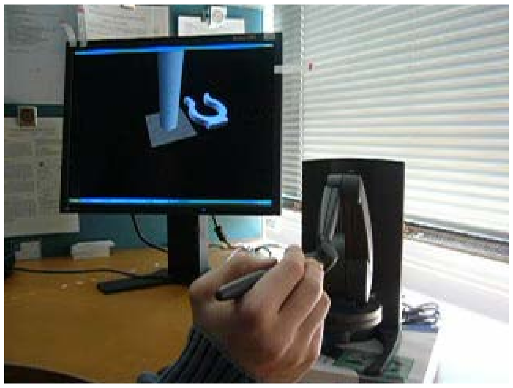


Figure 4.2: Interactive simulation of deformable objects.

rather late in a design process). Virtual humans are now starting to fill this need, but they are still limited by their lack of interactivity with the real world. While it is not too difficult to simulate a human moving in free space, what is actually asked for are:

- Active virtual humans (who interact with the surrounding virtual environment)
- Reactive virtual humans (who react to stress especially at work)
- Expressive human (who convey expressions)
- Cognitive human (able to behave with intelligence).

4.3.1 Avatars with hands

The first requirement is for virtual humans driven in an interactive way thanks to motion capture (avatars mimicking a human player). The main problem when controlling avatars in such a way is the difference between the real world and the simulated world. The virtual manikin to be driven can be morphologically different from the human ”controller” being tracked and this implies finding algorithms that recreate plausible movements enforcing defined constraints and goals. The second main difference between both worlds deals with the environment: the virtual human is to interact with a virtual product which has no real world counterpart (otherwise the method’s interest is lost). We must thus define a hybrid force/velocity strategy enabling to interact with the virtual environment while enforcing environment constraints (mainly non penetration).

Within the wide framework of virtual prototyping, most of the available manikins are handless and all the tasks that require grasping (a tool, an object or a support) do not take into account hand complexity and its implication in the whole manikin behavior. The need there consists in a hand that is:

- bio-mechanically similar to a human one (same degrees of freedom and limits)
- actuated via tendon by muscles models (Hill, Zajac ...)
- able of plastic deformation, when in contact with its environment
- able of reflex control and strategies, when grasping and manipulating

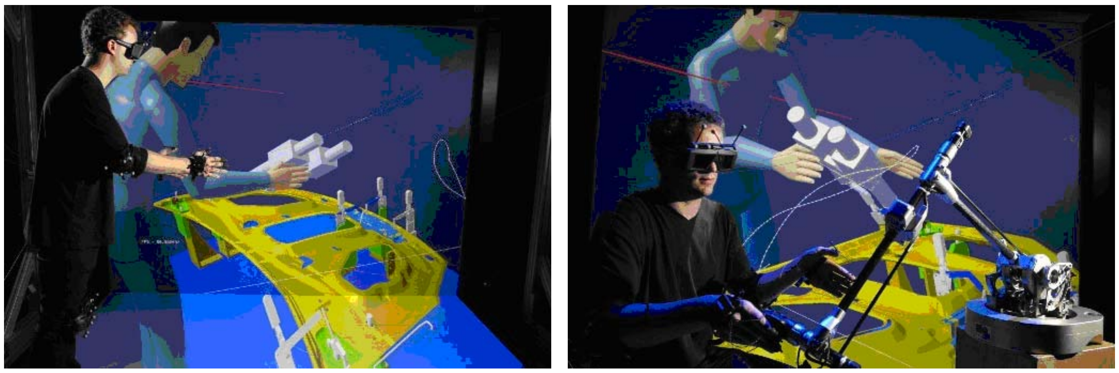


Figure 4.3: (a and b): Towards physically accurate virtual humans (Perf-RV2 project).

- easy to interactively control by an operator.

Finally, this virtual hand must be connected to a virtual manikin and controlled in a coordinated way.

The second step is to develop virtual humans that may be controlled at a higher level and this amplifies the difficulties previously encountered.

4.3.2 Robustness

In order to enrich the capabilities of virtual manikins, new controllers must take into account multiple non coplanar contacts (bilateral and unilateral with friction), not only for posture stability (or more precisely ensuring ”force closure” as for manipulation), but also in a more active way in order to compensate external perturbations or to execute a task which needs some preconfigured reactivity. Applying ”force closure” criteria, robustness can be obtained through an adapted COG (Center Of Gravity) position [63]. However, this does not give any indication on which value could take contact forces (or wrenches more generally). Some years ago in the scope of grasping and manipulation, the concept of ”perturbation closure” was introduced [41] that closely relies, after some simple mathematical manipulations, on the notion of impedance or stiffness control which have to be taken into account in the controller design.

At the research level, the main issue consists in finding the best compromise between the best rigorous formulation and the most efficient algorithm. More precisely, the difficulties come from nonlinear models, unilateral constraints leading to complementary conditions which are easily addressed in a linear context, but not in a nonlinear, even simply quadratic one. The compromise must be assessed on the basis of a ”quantitative/qualitative performance” versus ”computation charge” ratio.

4.3.3 Posture control

Starting from previous algorithm outputs as desired posture configuration, including a desired stiffness, it is essential to cope with dynamic control and stability. In other words, the question is: when the manikin is away from its equilibrium configuration, does there exist a control, and if yes, which is the best one for recovering the equilibrium configuration?

These questions could be answered separately, for example in building a stability region around the equilibrium configuration, but a ”simpler” solution may consist in addressing both of them simultaneously by looking for a possible control. Various solutions may be considered, from the very simple one consisting in coupling a PID control and a one step (one control shoot) optimization, to the more elegant NMPC (Nonlinear Model Predictive Control) formulation and its associated resolution methods such as collocation or multiple shooting [25]. Once again, the goal is a necessarily a ”quantitative/qualitative performance” versus ”computation charge” compromise. If no control is available for reaching the desired posture, another posture based on other supports must be determined.

4.3.4 Multi-objective control

Going further, posture control has to be completed in order that the manikin executes some useful tasks. This problem is related to task priority which can be solved in a simple way by choosing suitable weights in a criteria function. More elaborated treatment of this topic is based on projections: cinematic projections as proposed by [5], dynamic projections as proposed in [70] or virtual links which could be also defined as dynamic integrable projections as studied in [43]. To our knowledge and experience, the virtual link concept appears to be less fragile with regard to stability considerations (see for example [63]), but priority between several tasks is more difficult to introduce.

4.4 Representation of scenarios

When a Virtual Environment (VE) has been created implementing laws of physics and possibly embodying virtual humans, the VR developer still has to specify the general behavior of the VE components, in particular the virtual humans that are not directly controlled by the human player(s) (perform an assembling or maintenance task, pack some items in a container ...), and how these will react to the events that may occur. In other words, the scenario of the VR application has to be written. For standard applications, some very efficient tools are now available for generating scenarios. The way Virtools 4 (see below) supports the graphical edition of scenario through linking ”behavior building blocks” is particularly noteworthy.

When addressing virtual humans, things may become very complex. For example, Endorphin, a state-of-the-art software tool that is currently very popular in the gaming community, addresses the problem of animating virtual characters with the following set of behaviors:

- Hand and arm behaviors
- Arms Crossed On Chest
- Arms Raised Above Head
- Arms Wide Of Head
- Arms Windmill
- Arms Zombie
- Hands Covering Face
- Hands From Behind Back
- Hands Protecting Groin

- Hands Reach And Look At
- Leg behaviors
- Legs Kick
- Legs Reach
- Legs Straighten
- Whole body behaviors
- Balance
- Balance With Props
- Body Foetal
- Catch Fall
- Fall Back, Twist And Catch Fall
- Fall Back, Twist And Cover Face
- Fall Back, Twist With Passive Arms
- Jump
- Jump And Dive
- Land And Crouch
- Stagger
- Tackle
- Writhe
- Writhe In Mid-Air
- Other behaviors
- Body Stiffness
- Body Damping
- Whole Body Stiffness
- Hold.

Implementing higher-level behaviors according to the objectives of the considered virtual character is a current research topic. This becomes even more crucial with virtual manikins currently being developed in laboratories (see the previous section) that attempt to closely interact with their surrounding virtual environment and to sustain tiredness, physical damages or health problems.

4.5 Existing tools

Basically, a Virtual Reality (VR) application needs a set of data that describes the Virtual Environment (VE) with which the human player(s) interacts. To manage this human / VE interaction, the application developer must then specify the VE behavior according to the various events that may occur. Finally, the resulting application have to run on a virtual reality system that is sometimes very sophisticated, embodying numerous input and output devices to support multi-modal interactions and possibly operating in a networked environment. A VR application thus requires:

- import functions to convert the VE data from their native format (for example CAD) into the considered VR format optimized for real-time efficiency
- a development toolkit to define the behavior of the VR application
- a runtime (VR player).

At the moment, there are numerous freeware or commercially available software products that fulfill these functions. A number of the most significant ones are briefly described below.

When developing a VR application, interfacing specialized input/output devices is often an inescapable task and some dedicated tools appeared very early.

Input device interface libraries
A number of libraries are available to simplify the management of the various input/output devices encountered in VR systems. Some are commercial products such as trackd (VRCO, http://www.vrco.com/trackd/Overviewtrackd.html), others are open source like VRPN (University of North Carolina Chapel Hill) or OpenTracker (University of Vienna). Recent VR development toolkits generally address this question and provide built-in drivers for device interface.

Several complete VR development tools also emerged from the first VR labs to use either with standard graphic software or as standalone.

CAVElib
CAVElib is a C programming interface that increases the VR capabilities of Performer, OpenInventor or other OpenGL renderers. It especially manages the important features of VR systems in order to assist in building virtual reality applications: display configurations, multi-pipe parallel rendering, stereo and tracking. Originally designed and developed at EVL, University of Illinois at Chicago since 1992, CAVElib has been used in a number of VR applications and is available in a commercial version from VRCO. See http://www.vrco.com/CAVELib/OverviewCAVELib.html.

VR Juggler
VR Juggler is an open-source VR application development framework designed by the Iowa State University's Virtual Reality Applications Center. It provides a set of application programming suites (APIs) that deals with interface management, display surfaces, tracking, navigation, graphics and rendering techniques. The developed applications are flexible enough to operate under any circumstances independently of the connected I/O devices and the computer platform. The VR Juggler Suite includes a virtual application, a device management system (for local or remote access to I/O devices), a standalone generic math template library, a portable runtime (providing cross-platform thread, socket and serial port primitives), a simple sound abstraction, a

distributed model view-controller implementation and an XML-based configuration system with multivariate types.

A natural approach was to implement OpenGL modules extending the real-time capabilities of this graphic language.

OpenGL Performer
OpenGL Performer (http://www.sgi.com/products/software/performer/) is a programming interface helping developers to create real-time simulation and other performance-oriented 3D graphics applications. It features a number of capabilities oriented towards complex applications such as visual simulation, manufacturing, virtual reality systems, scientific visualization, interactivity and computer-aided design. It can also manage multi-processor and multi-graphics pipelines when big amounts of processing power are needed. OpenGL Performer has been built on top of the OpenGL standard graphics library and combines Ansi C and C++. The suite can be used for application development on a variety of operating environments such as SGI IRIX, Linux and MS Windows (2000 and XP).

OpenSG
OpenSG (http://www.opensg.org/) is a portable scene-graph system to support real-time graphics rendering, in particular for virtual reality applications. Based on OpenGL, it was developed following Open Source principles and can be freely used. It runs on IRIX, Windows and Linux. As a scene-graph, it does not constitute a full VR system, but only an efficient visualization engine that handles all the scene primitives and enables multithreaded asynchronous scene-graph manipulations.

OpenSceneGraph
The OpenSceneGraph (http://www.openscenegraph.org/) is an open source high performance 3D graphics toolkit used by application developers in fields such as visual simulation, games, virtual reality, scientific visualization and modelling. Written entirely in Standard C++ and OpenGL it runs on all Windows platforms, OSX, GNU/Linux, IRIX, Solaris and FreeBSD operating systems. As OpenSG, it only addresses a part of VR development.

OpenSceneGraph may be also associated to OSGEdit, a tool to edit 3D scenes for the OSG library. OSGEdit is only a composer (not a modeller) whose purpose is to compose complex scenes based on individual models using the OSG library. It is focused on making scenes for use in OSG-based programs.

Several well-known products in the field of visualization and 3D interfaces were also extended to feature virtual reality capabilities.

WorldToolKit
WorldToolKit (from Sense8, http://www.sense8.com/) is a software tool addressing the development of interactive 3D applications for scientific and commercial use. It incorporates a complete set of function libraries and tools required to create, manage and commercialize applications. It features an Application Programmers' Interface (API) that allows its user to prototype, develop and configure applications. This API gathering about 1000 C-code functions is organized in classes dedicated to geometric objects, viewpoints, sensors, paths, lighting and other features (like rendering).

A development environment (WorldUp) is also available to facilitate the creation of standard VR applications. WorldToolKit provides options for immersive displays and interface devices (Helmet Mounted Displays, trackers, etc). It also supports networked distributed simulations and Sense8 proposes a client/server based networking solution named World2World. WorldToolKit is portable across various platforms ensuring compatibility with most operating systems such as Windows, Linux and Sun.

Open Inventor

Open Inventor (http://www.tgs.com/) is a powerful 3D graphics suite that aims at simplifying and accelerating the development of cross platform (IRIX, Solaris, HP-UX, AIX, Linux, and Windows) 3D graphics applications in C++, .NET and Java. Open Inventor is able to handle large data sets and real-time interactions. Based on OpenGL for the graphics rendering, it also embodies several interesting functions for VR like multithreading, remote rendering, stereo viewing, NURBS, large model visualization, collision detection, HTML image maps, 3D textures, VRML compatibility and special graphical effects libraries.

In the TGS product line, 3D visualization and volume modelling are performed by Amira (http://www.tgs.com/pro_div/amira_main.htm) which enables working on very large data sets in real time.

Finally some development tools are totally dedicated to virtual reality applications.

Virtools

The Virtools company which is now a subsidiary of Dassault Systèmes has developed one of the most complete development and deployment platform for VR applications. It includes tools for capturing 3D data from an extended number of sources, for developing interactive applications and for deploying on various platforms and in networked environments. The strong features of the Virtools product line are (1) its proximity with Catia which allows an easy importation of CAD data (its internal 3D XML 3D format is compatible with other Dassault Systèmes software tools), (2) its scenario editor, which offers an intuitive user interface based on ”behavior building blocks” and ease the creation of complex interactive application and (3) the large number of available module for interfacing VR devices, implementing physics simulation or creating intelligent (AI-based) virtual actors. For more details, see http://www.virtools.com/solutions/products/virtools_4.asp.

Chapter 5

Net-WMS Methodology and Architecture

5.1 Constraint Programming Approach to Bin Packing Problems

As shown in this state-of-the-art, the bin packing problem is a core problem of supply chain management. Industrial problems such as container loading or pallet loading problems add however some additional constraints and requirements that need be handled to tackle the real problems.

The main technical approach developed by the Net-WMS project to solve these problems is based on constraint programming. In particular, we aim at making significant advances on the design of, and experimentation with:

- non-elementary global geometrical constraints with efficient filtering algorithms [11];
- bin selection and placement heuristics taking benefit of constraint handling;
- optimization constraints in this context.

As the goal is to efficiently solve industrial problems in real size, the other methods, and their hybridation with CP, mentioned in the state-of-the-art about linear programming and local search, will be considered as complementary techniques for improving the efficiency of our prototypes developed for the study cases.

5.2 Knowledge Representation with Packing Business Rules

The Net-WMS project aims to design the PackML markup language for defining packing problems in a declarative manner, and solving them efficiently by constraint programming. To this end, three layers of languages are considered [53]:

1. geometrical constraints, including powerful global constraints like [11];
2. placement constraints, like the Region Connection Calculus [61];
3. placement business rules.

While the translation from placement constraints to geometrical constraints is quite clear, the translation from packing business rules to efficient constraint programs is one main challenge addressed by the project.

5.3 Optimization with User-Interaction in Virtual Reality

Based on the brief state-of-the-art addressing enabling Virtual Reality (VR) technologies, the VR part of the Net-WMS project may be settled in four main points:

- Model of the packing problem.

  The key task consists in defining the considered processes through a precise and practical specification of their crucial components, in particular:

  - the scope of the activities taken into account (arrangement of items, loading, transport, unloading ...)
  - the relevant level (or levels) of detail
  - the relevant parameters (shape, weight, center of gravity ...)
  - the dynamic behavior models whose implementation may provide a real benefit (deformations, dry friction ...)
  - the smooth integration within the existing business software environments (interoperability).

  This stresses the need for an early specification of the useful and achievable functions for future WMS software.

- 3D graphical interface allowing users to interact with the above model.

  While developing an efficient 3D user interface is a necessary step for any VR application, some specific care is required here to identify the future users of the system (planners, workers ...) and, most importantly, their particular needs.

- Interaction with the optimization functionalities through VR.

  This is clearly one of the interesting parts of the VR involvement in this project. It implies displaying the output of the optimization algorithms in a way that allows the user to understand it, in order to combine the optimization capabilities with human skill.

- Interactive simulation taking into account the dynamic behavior of the packaged items.

5.4 Networked Architecture

All components developed in the Net-WMS project will be designed as middleware components in a networked J2EE architecture. This means that the resources and services concerning

- the packing business rules,
- the packing solvers,
- the virtual reality resources and services,

will be made available across the network with different user front-ends.

Bibliography

[1] R. Alvarez-Valdes, F. Parreno, and J.M. Tamarit. A tabu search algorithm for the pallet loading problem. *OR Spectrum*, 27(1):43–61, 2005.

[2] J. Amsden, D.J. Berg, K. Brown, G. Craig, G. Hester, P. Jakab, D. Pitt, R. Stinehour, and M. Weitzel. *An Overview of J2EE with IBM WebSphere*. Prentice Hall, 2004.

[3] M. Anitescu and F. Potra. Formulating dynamic multi-rigid-body contact problems with friction as solvable linear complementarity problems, 1997.

[4] Y. Azar and O. Regev. On-line bin-stretching. *Theoretical Computer Science*, 268:17–41, 2001.

[5] Paolo Baerlocher. *Inverse kinematics techniques of the interactive posture control of articulated figures*. PhD thesis, Lausanne, 2001.

[6] David Baraff. Curved surfaces and coherence for non-penetrating rigid body simulation. In *SIGGRAPH '90: Proceedings of the 17th annual conference on Computer graphics and interactive techniques*, pages 19–28, New York, NY, USA, 1990. ACM Press.

[7] David Baraff. Fast contact force computation for nonpenetrating rigid bodies. In *SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, pages 23–34, New York, NY, USA, 1994. ACM Press.

[8] F. W. Barnes. Packing the maximum number of $m \times n$ tiles in a $p \times q$ rectangle. *Discrete Mathematics*, 26:93–100, 1979.

[9] R. Barták, T. Müller, and H. Rudová. A new approach to modelling and solving minimal perturbation problems. In Springer Verlag, editor, *Recent Advances in Constraints, 2003*, volume LNAI 3010, 2004.

[10] N. Beldiceanu. Global constraints as graph properties on a structured network of elementary constraints of the same type. In R. Dechter, editor, *Principles and Practice of Constraint Programming (CP'2000)*, volume 1894 of *LNCS*, pages 52–66. Springer-Verlag, 2000.

[11] N. Beldiceanu, M. Carlsson, E. Poder, R. Sadek, and C. Truchet. A generic geometrical constraint kernel in space and time for handling polymorphic k-dimensional objects. In *Proceedings of the 13th Conference on Pinrciples and Practice of Constraint Programming CP'07*, Lecture Notes in Computer Science, Providence, MA, USA, September 2007. Springer-Verlag.

[12] E. Burke, R. Hellier, G. Kendall, and G. Whitwell. A new bottom-left-fill heuristic algorithm for the two-dimensional irregular packing problem. *Operations Research*, 54(3):587–601, 2006.

[13] E. Burke, R. Hellier, G. Kendall, and G. Whitwell. Complete and robust no-fit polygon generation for the irregular stock cutting problem. *European Journal of Operational Research*, 179:27–49, 2007.

[14] J Canny. Collision detection for moving polyhedra. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(2):200–209, 1986.

[15] J. Carlier, F. Clautiaux, and A. Moukrim. New reduction procedures and lower bounds for the two-dimensional bin packing problem with fixed orientation. *Computers & Operations Research*, 34:2223–2250, 2007.

[16] F. Clautiaux, J. Carlier, and A. Moukrim. A new exact method for the two-dimensional bin-packing problem with fixed orientation. *Operations Research Letters*. To appear.

[17] F. Clautiaux, J. Carlier, and A. Moukrim. A new exact method for the two-dimensional orthogonal packing problem. *European Journal of Operational Research*. To appear.

[18] F. Clautiaux, A. Jouglet, J. Carlier, and A. Moukrim. A new constraint programming approach for the orthogonal packing problem. *Computers & Operations Research*. To appear.

[19] F. Clautiaux, A. Jouglet, and J. El Hayek. A new lower bound for the non-oriented two-dimensional bin-packing problem. *Operations Research Letters*. To appear.

[20] E.G. Coffman, M.R. Garey, and D.S. Johnson. Approximation algorithms for bin packing: a survey. In D. Hochbaum, editor, *Approximation Algorithms for NP-Hard Problems*, pages 46–93. PWS pub., Boston, 1996.

[21] E.G. Coffman and G.S. Lueker. Approximation algorithms for extensible bin packing. *Journal of Scheduling*, 9(1):63–69, 2006.

[22] János Csirik, Davis S. Johnson, Claire Kenyon, James B. Orlin, Peter W. Shor, and Richard Weber. On the sum-of-square algorithm for bin packing. *Journal of the ACM*, 53(1):1–65, January 2007.

[23] K.M. Daniels and V.J. Milenkovic. Column-based strip packing using ordered and compliant containment. In *Proc. of the first ACM workshop on Applied Computational Geometry*. ACM Press, May 1996.

[24] A.P. Davies and E.E. Bischoff. Weight distribution considerations in container loading. *European Journal of Operational Research*, 114:509–527, 1999.

[25] Moritz Diehl, Hans Georg Bock, Holger Diedam, and Pierre-Brice Wieber. Fast direct multiple shooting algorithms for optimal robot control. *LNCIS*, 2006.

[26] K. A. Dowsland. The three-dimensional pallet chart: An analysis of the factors affecting the set of feasible layouts for a class of two- dimensional packing problems. *Journal of the Operational Research Society*, 35:895–905, 1984.

[27] K. A. Dowsland. Determining an upper bound for a class of rectangular packing problems. *Comput. Oper. Res.*, 12:201–205, 1985.

[28] K.A. Dowsland and W.B. Dowsland. Solution approaches to irregular nesting problems. *European Journal of Operational Research*, 85:506–521, 1995.

[29] L. Epstein. Bin stretching revisited. *Acta Informatica*, 39(2):97–117, 2003.

[30] Michel Journet et al. *Le Guide de la Logistique, Editions DALIAN*. 2005.

[31] François Fages, Julian Fowler, and Thierry Sola. A reactive constraint logic programming scheme. In Leon Sterling, editor, *Proc. International Conference on Logic Programming ICLP'95*, Tokyo, 1995. MIT Press.

[32] François Fages, Julian Fowler, and Thierry Sola. Experiments in reactive constraint logic programming. *Journal of Logic Programming*, 37:1-3:185–212, October 1998.

[33] François Fages, Sylvain Soliman, and Rémi Coolen. CLPGUI: a generic graphical user interface for constraint logic programming. *Journal of Constraints, Special Issue on User-Interaction in Constraint Satisfaction*, 9(4):241–262, October 2004.

[34] G. Fasano. A mip approach for some practical packing problems: Balancing constraints and tetris-like items. *4OR: A Quarterly Journal of Operations Research*, 2(2):161–174, 2004.

[35] Erich Friedman. Packing unit squares in squares. A survey and new results. *Electronic Journal of Combinatorics*, 7, 2005. *http://www.stetson.edu/ efriedma/papers/squares/squares.html*.

[36] M.R. Garey and D.S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. Freeman, New York, 1979.

[37] S. Gokul. *J2EE: What it is and what it's not*. Prentice Hall, 2002.

[38] S. Gottschalk, M. C. Lin, and D. Manocha. OBBTree: a hierarchical structure for rapid interference detection. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 171–180, New York, NY, USA, 1996. ACM Press.

[39] Naga K. Govindaraju, Stephane Redon, Ming C. Lin, and Dinesh Manocha. CULLIDE: interactive collision detection between complex models in large environments using graphics hardware. In *HWWS '03: Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware*, pages 25–32, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[40] Business Rules Group. Business rules manifesto, 2003. `http://www.businessrulesgroup.org/brmanifesto.htm`.

[41] M.-K. Hor and S.-C. Wu. On the grasping stability and optimality under external perturbations. pages 509–526, 1999.

[42] H. Isermann. Obere schranken für die lösung des zweidimensionalen packproblems auf der basis struktureller identitäten. In Fandel and Gehring, editors, *Operations Research – Beiträge zur quantitativen Wirtschaftforschung*, pages 341–348. Springer-Verlag, 1991.

[43] L.D. Joly. *Commande hybride position/force pour la téléopération: une approche basée sur des analogies*. PhD thesis, Université de Paris 6, 1997.

[44] R. Keber. Stauraumprobleme bei stückguttransporten. Technical Report Heft 17, Wissenschaftliche Berichte des Institutes für Fördertechnik der Universität Karlsruhe, 1985.

[45] Young J. Kim, Miguel A. Otaduy, Ming C. Lin, and Dinesh Manocha. Fast penetration depth computation for physically-based animation. In *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 23–31, New York, NY, USA, 2002. ACM Press.

[46] R.E. Korf. Optimal rectangle packing: Initial results. In *Proc. of the 13th International Conference on Automated Planning and Scheduling (ICAPS-2003)*, pages 287–295, 2003.

[47] R.E. Korf. Optimal rectangle packing: New results. In *Proc. of the 14th International Conference on Automated Planning and Scheduling (ICAPS-2004)*, pages 142–149, 2004.

[48] R. LeMaster. Alex – an expert system for truck loading. In *Proc. of the 3rd Int. Conf. on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, volume 2, pages 638–644, 1990.

[49] M. C. Lin and S. Gottschalk. Collision detection between geometric models: a survey. In *IMA Conference on Mathematics of Surfaces*, pages 37–56, San Diego, USA, 1998.

[50] A. Lodi, S. Martello, and D. Vigo. Recent advances on two-dimensional bin packing problems. *Discrete Applied Mathematics*, 123:379–396, 2002.

[51] P. Lotstedt. Numerical simulation of time-dependent contact and friction problems in rigid body mechanics. *SIAM Journal on Scientific and Statistical Computing*, 5(2):370–393, 1984.

[52] P. Lotstedt. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. *International Journal of Numerical Methods in Engineering*, 1996.

[53] Julien Martin and François Fages. From business rules to constraint programs in wharehouse management systems. In *Proceedings of the 13th Conference on Pinrciples and Practice of Constraint Programming CP'07*, Lecture Notes in Computer Science, Providence, MA, USA, September 2007. Springer-Verlag.

[54] K. Mathur. An integer-programming-based heuristic for the balanced loading problem. *Operations Research Letters*, 22(1):19–25, 1998.

[55] Sun Microsystems. *Simplified Guide to the Java 2 Platform, Enterprise Edition*. Sun Microsystems.

[56] C.J. Ong. The Gilbert-Johnson-Keerthi distance algorithm: A fast version for incremental motions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1997.

[57] D. Pisinger, E. den Boef, J. Korst, S. Martello, and D. Vigo. Algorithms for general and robot-packable variants of the threedimensional bin packing problem. *ACM Transactions on Mathematical Software*. To appear.

[58] D. Pisinger and M. Sigurd. Using decomposition techniques and constraint programming for solving the two-dimensional bin packing problem. *INFORMS Journal On Computing*, 19(1):36–51, 2007.

[59] Rolf G. Poluha. *Application of the SCOR Model in Supply Chain Management, Youngstown, ISBN 1-934043-10-9*. 2006.

[60] J. Puchinger and G.R. Raidl. Models and algorithms for three-stage two-dimensional bin packing. *European Journal of Operational Research*. To appear.

[61] David A. Randell, Zhan Cui, and Anthony Cohn. A spatial logic based on regions and connection. In Bernhard Nebel, Charles Rich, and William Swartout, editors, *KR'92. Principles of Knowledge Representation and Reasoning: Proceedings of the Third International Conference*, pages 165–176. Morgan Kaufmann, San Mateo, California, 1992.

[62] S. Redon and S. Coquillart. Interval analysis for computer graphics. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000.

[63] A. Rennuit. *Contribution au contrôle des humains virtuels interactifs*. PhD thesis, Ecole Centrale de Nantes, Spécialité Génie Mécanique, CEA Fontenay aux Roses, 2006.

[64] John Alan Roach. The rectangle placement language. In *DAC '84: Proceedings of the 21st conference on Design automation*, pages 405–411, Piscataway, NJ, USA, 1984. IEEE Press.

[65] E. Roman, R.P. Sriganesh, and G. Brose. *Mastering Enterprise JavaBeans*. Wiley Computer Publishing, 2002.

[66] Michel Roux and Tong Liu. *Optimisez votre plate-forme logistique, Editions d'Organisation, ISBN : 2-7081-3133-8*. 2004.

[67] Hani El Sakkout, Thomas Richards, and Mark Wallace. Minimal perturbation in dynamic scheduling. In John Wiley & Sons, editor, *13th European Conference on Artificial Intelligence ECAI-98*, 1998.

[68] G. Scheithauer and J. Terno. Modelling of packing problems. *Optimization*, 28:63–84, 1993.

[69] G. Scheithauer and J. Terno. The G4-Heuristic for the Pallet Loading P roblem. *European Journal of Operational Research*, 46:511–522, 1996.

[70] L. Sentis and O. Khatib. Control of free-floating humanoid robots through task prioritization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1730–1735, Barcelona, Spain, 2005.

[71] P. Shaw. A constraint for bin packing. In *Principles and Practice of Constraint Programming – CP2004 (Springer LNCS 3258)*, pages 648–662, 2004.

[72] J.M. Snyder. Interval analysis for computer graphics. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 121–130, New York, NY, USA, 1992. ACM Press.

[73] J.R. Wodziak and G.M. Fadel. Packing and optimizing the center of gravity location using a genetic algorithm, 1994. Unpublished manuscript.