Design of a Flexible Graphic Visualizer for Flowshops Scheduling

Rodolfo Ruiz Nangusé, Larysa Burtseva, and Gabriel A. López Morteo

Engineering Institute of Autonomous University of Baja California, Calle de la Normal S/N, 21280, Mexicali, BC, Mexico
{rodolfor, lpb, galopez}@iing.mxl.uabc.mx

Abstract. This paper describes the Flexible Graphic Visualizer, designed for the graphic interpretation of the results of flowshop scheduling algorithms obtained in a computational system. The visualizer is intended for the understanding of the data model generated by a scheduling algorithm. The visualizer interface integrates different visualization and interaction techniques for a deep and interactive exploration of the information represented in a visual manner. Due to the capacity of the visualizer to represent graphically data of any scheduling context, this system is useful to visualize complex data from flowshop scheduling algorithms both on production environments or research.

Keywords: visualization, scheduling, flowshop, Gantt chart, visualization techniques.

1 Introduction

This paper describes flexible visualization software for graphic interpretation of the results of the execution of algorithms for flowshop scheduling.

The first work dedicated to job scheduling, was published in 1950's by S. M. Johson [1]. Since those times, the problem attracted the attention of many researches because the theory complexity, and diverse of practical's applications [2, 3]. Due to this complexity, the analysis and execution of scheduling algorithms require the development of computational systems, such as visualization tools, that help in the interpretation of the algorithms' results. With the help of these tools, it is possible to make clear different schedules effects, such as "bottle necks", machines employment, homogeneity, etc.

When a specific scheduling algorithm is implemented in industrial environments, it is important for the operators to have adequate representations of the data, showing not just the schedules but related information, in a comprehensible and comfortable form, to support the organization of the productive process. The information related to the schedules are, allocation of jobs to the resources, monitoring of jobs' dates to machines, setup times in machines, WIP, interruptions of machines for blocking, etc.

The visualization systems or visualizers available at this time, both commercial and particular, were designed for certain groups of problems according to the interests of their users. Taking this in account, the properties and behaviors of visualizers allow

© A. Gelbukh, M. Adiba. (Eds.)
Advances in Computer Science and Artificial Intelligence.
Research in Computing Science 39, 2008, pp. 263-276

Received 01/07/08
Accepted 01/08/08
Final version 11/09/08

classifying them in three categories: *layout visualizers*, *automatic visualizers* and *enterprises solutions visualizers*.

Layout visualizer provides the user with a set of tools, such as lines and forms, to generate graphs from the results of its scheduling algorithms in a manual way [4, 5].

Automatics visualizers have an interface for interaction with the user; involve programming to generate graphics according to a wanted context. They provide diverse tools for the exploration of the data [6, 7, 8].

Enterprises solutions visualizers are integrated in packages of applications that offer to the companies' solutions to their problems in diverse areas, such as the management of departments, the planning of the production and projects, etc. The features for these visualizers are very general due to their tendency of embracing diverse areas. The specialization and robustness of this kind of visualizers are in proportion with the cost and the maker's prestige [9, 10].

Due to the complexity in the generation of calendars, standard free software doesn't exist for charting the results of specialized algorithms in scheduling. The commercial visualizers are oriented to enterprise companies at very high cost [7, 9, 10]. The visualizers developed for research is oriented to particular uses, for example, [11, 12].

The investigation of algorithms and visualizations of results for complex models of scheduling imply a variety of elements to be represented in the same chart. Otherwise, the implementation of visualizer in the factory requires diverse focuses in a schedule with the purpose of acquiring valid conclusions about the productive process.

This article describes the design of a flexible graphic visualizer (VGF) that achieves all requirements above mentioned. It is employed as an independent component for the system PLARETF, developed for execution and analysis of flowshops scheduling algorithms [13, 14].

The rest of the paper is organized as follows: Section 2 explains the definitions and intrinsic notations in a scheduling problem, shows up the visualization of problem results using a Gantt chart. The Section 3 describes the state of the art for the knowledge acquisition from a dataset that are studied in a specific context. The Section 4 provides the design of VGF. Presenting the scheduling knowledge model and the visualization techniques identifies according to the characteristics of the scheduling; later is detailed the elements of VGF interface in which are reflected the visualization techniques studied and the way we relate them to provide many perspectives and details of the information. In another section the characteristics of VGF flexible input format that allows represented graphically different models of shops scheduling is showed. Also, the VGF architecture showing the outline of its elements and the communication that exists among them to provide the required functionalities are presented. Lastly, section 5 contains the conclusions of the work.

2 Scheduling Problem

Different aspects of scheduling are examined in scientific literature [15, 16]. The problems are formulated in terms of "jobs" which are processed in "machines" with various constraints. The jobs flow pattern of the machines defines the shop model. A conveyor with m successive operations represents a simple flowshop (FS). If for the execution of at least operation are several machines available, such shop called as flexible flowshop with m stages (FFS), if the machines in a group are identical, and hybrid flowshop (HFS), if these they are uniform (machines with different speeds) or not related (different machines). The Fig. 1 show the resources system in a FFS with m stages, each of which contains m_i parallel machines, $i = 1, \ldots, m$.

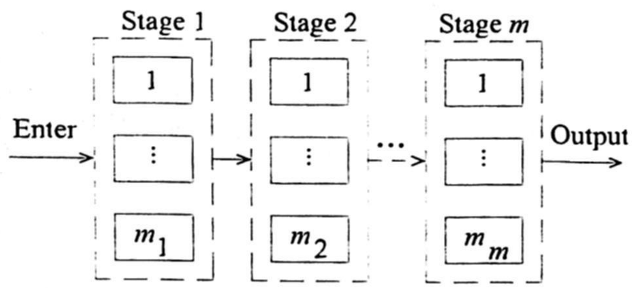

Fig. 1. Flexible (hybrid) flowshop with m stages.

With each one job j, $j = 1, \ldots, n$, several data are associated, as processing time p_{ilj} in the machine i of stage l, arrival instant r_j in shop, due day d_j that represents the dates the job j is promised to the customer, weight w_j which is basically a priority factor of job related to other jobs in the system, setup time s_{iljk} of the machine i to process the job k after the job j. the model specified other properties and restrictions as set M_j of eligible machines for job j when in the machine environment are several machines in parallel, machines blocking which may occur in flowshops when it has limited buffer between two successive machines for work in process (WIP) between two successive machines, etc.

The scheduling aim is to satisfy criteria. The most common optimization criterion is to minimize the jobs set processing time (makespan) since usually implies a high utilization of the machines. Others criteria exist, such as minimize the jobs' finalization time, considering jobs arrival at the system (release times), finalization compromise dates (due dates), priorities (weights), precedence relationships, the number of jobs retarded, among others.

Here an example is presented. It is necessary to process 6 jobs in a HFS of two stages with a machine in the first stage and two machines in the second. All the jobs consist of two successive operations. The first operation carries out by unique machine of the first stage. To execute the works 1, 2, 3 is eligible the machine 1 of the second stage and for the other three works the machine 2 is eligible. The jobs release times are $r_j = 0$, $j = 1, \ldots, 6$. The Table 1 shows the jobs' processing time for the example. The non eligible machines for a definitive job are marked with the sign "-".

Table 1. Jobs processing time

Job *j*	Machina 0	Machina 1	Machina 2
1	1	4	-
2	5	9	-
3	3	6	-
4	5	-	5
5	4	-	10
6	8	-	7

The permutation π^* = (5, 2, 6, 3, 4, 1) corresponds to optimal schedule with $C_{\max} = 30$. The Fig. 2 shows the problem solution in Gantt chart form for the permutation π^*.

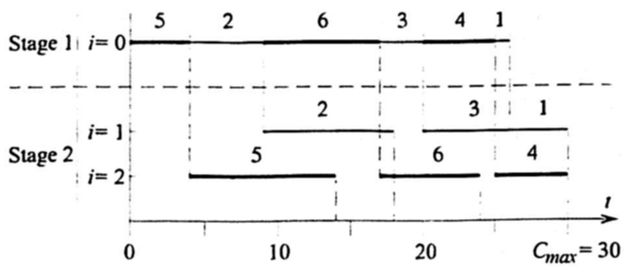

Fig. 2. Gantt chart for the optimal permutation π^* = (5, 2, 6, 3, 4, 1)

As shown in the example, the Gantt chart clarifies the visualization of the assignment and the movement of jobs on the machines, timing of each job, and idle times of each machine. In a similar way, other properties and restrictions of the shop are indicated in the chart. The Gantt chart is applicable for any kind of shop: flowshops, openshop, jobshop, therefore they are used in different scheduling visualizers [6, 7, 8, 9, 10].

The visualization of scheduling shops has its intrinsic complexity due to variations in:

- Length of Gantt chart that vary from short periods of time to long periods, until thousands time units;
- Models of resources as number of stages in shop, number of machines for stage, type of flow of jobs.
- Concepts and values to be displayed, such as makespan, setup times of the machines, release times, WIP, and blocking.
- Necessity of obtaining and visualizing the information for demand.

3 Acquisition of the Knowledge through the Visualization

3.1 Knowledge Acquisition Stages

The visualization is the creation of a mental image from an abstract concept [17], to perceive the characteristics of the data examined. Diverse approaches exist about the knowledge acquisition, locating the visualization as a layer between the human and the information [18]. In publications referred below, authors analyzed how the knowledge was represented to visual form from raw data. According to [19], the knowledge acquiring process is composed by four conceptual elements that evolve in the time: those elements are involved in an understanding context.

In the lowest part of the model, the raw data are stored as a characters set. In the following level metadata are added, to bind the raw data to particular context, resulting in what is denominated as information.

Until this point the raw data and the information are denominated as *producers* because they produce the information about a context. In a higher level, the information interacts with a human. The data are interpreted when being placed in a context. Based on the experience, the knowledge has achieved. In this point, the knowledge and the information are denominated as *consumers*. Finally become a concept called *wisdom*, which is achieved using and combining the knowledge in different forms and situations inside the individual's context.

Another approach is proposed in [20]. In the visualization process there are four basic phases that feedback between the last in the first. The first phase consists on the recollection and storage of the data, the second phase pre-processes and it transforms the data in information according to the individual's environment; the third phase deploys the graphically information that activates the visual and cognitive human systems. An analyst interacts with the information in three aspects. The first one is the data recollection from any physical environment where they are stored, and a social environment that determines their interpretation. Aspects as the exploration and manipulation form the second aspects where the data are been understanding for the human, and the third aspect is about the visual device, the manner which the information is represented to the user in visual form.

In [21] described the transformation phases of a dataset into visual forms for human collectors. In the first phase, the raw data are converting in tables of data through of metadata aggregation. The following phase is to add a visual structure to the data tables producing a visual representation. This is achieved through visual mappings. The last phase involves the visual mappings interpretation to show them to the user through a visual device. The user feedback the system through the change of parameters that control them the three phases of transformation above mentioned.

The visualization concept approach development by Card [21] was used as conceptual base for the VGF design, and is described in section 4.

3.2 Visual Exploration of Data

Three processes compose the exploration of data in a visual way: panorama, zoom and filtrate [22]. First, the user needs to get an early panorama of the data, where patterns are identified and focused in one or more of them. To analyze patterns, it is needed to access data details. The user identifies a data subset, deepens the panorama through zoom and filtration process, decomposing the subset and exploring its elements to obtain details. It is important stress that the visualization technology not only provides the bases of the techniques for the three processes of the data exploration but it is also the bridge among them.

3.3 Visualization Techniques

Below the techniques applied for the interactive visualization charts are described.

Keim [22] classifies the visualization techniques according to three approaches:

- Characteristics of the data.
- Technique used for visualization.
- Technique of interaction with the data.

The data characteristics specified the data types represented: one-dimensional, bi-dimensional data, text, hypertext, hierarchy, graph, algorithm, and software.

In [23] a deep analysis is made about the visualization techniques and the context in what are appropriate: 2D and 3D graphics, geometric transformations, iconic representations, density diagrams of pixels (stacked diagrams are an example of them.) Also in the same paper authors analyzed how the data can be explored and manipulated.

4 VGF Design

The VGF described in this article, is an application developed for the necessity of representing graphically the results from PLARETF [13] system. The VGF system has a flexible input format, and allows the graphical representation of any shop type, including openshop and jobshop, with their characteristics and restrictions.

The agile understanding of the model is reached through the application of interaction technologies. Below some visualization aspects are described in the scheduling context.

4.1 Scheduling Results Knowledge Model through VGF

The model of understanding through visualization, presented in fig. 3, describes how the data generated by a scheduling algorithm, are understood by an analyst. Three transformation processes and a feedback process compose it.

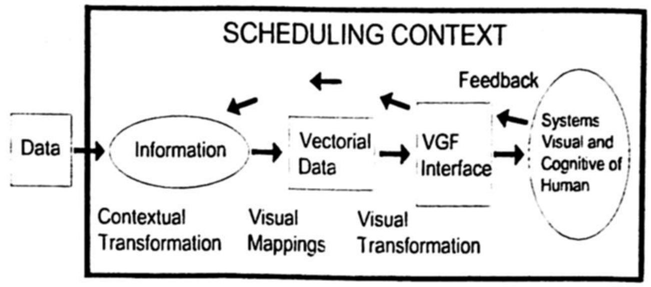

Fig. 3. VGF scheduling results knowledge model

In the process of *contextual transformation*, output data of an algorithm, stored in a text file in the input format of VGF are interpreted in the scheduling context. Thus, the raw data has become information. The VGF component, transforms the information from scalar to the vector adding *visual mappings*.

Later, the VGF component *transforms graphically* the vector information through the interface, showing the algorithms results in graphical form. In addition, the VGF interface offers a collection of tools that allows user interaction with the information. In this point, the *feedback* process facilitates the access to the calendars data in plain text, to manipulate the information and to explore it in detail, activating the cognitive and visual systems of the user. The transformation and processes feedback are inside the scheduling context.

4.2 Visualization Techniques

The VGF used the following visualization techniques [22].

– **Data representations:**

The system uses text to define labels and comments. The jobs processing dates are defined as two-dimensional data of text; it is an array whose each line describes the job timing. The data involved in the shop model representation are hierarchical, in tree form whose root is the shop, the machines occupy a second level, and the works are the leaves.

– **Visualization Techniques:**

As the main representation for the algorithms results provided by PLARETF, we use Gantt charts for the representation of scheduling characteristics and restrictions. The Gantt charts allow visualizing, in an agile way, the jobs processing order in the shop, indicating the machine used, and the beginning and end instants for each operation of the jobs.

The diagram radar view shows a scaled representation of the diagram of Gantt to give the user a total perspective of the image. This is usefully when an original schedule is very extensive and doesn't spread out entirety in the screen. This is important when, for example, a schedule has a big number of stages, machines, and jobs; big values of times; and numerous restrictions and characteristics. This kind of diagrams offers the possibility to browse the data and to observe in detail the characteristics of a subset of data.

The third visualization technique implemented is a tree diagram, which takes advantage of the hierarchical characteristics of the data and it deploys to the tree a directory of elements whose three levels are the shop, the machines and the jobs. This kind of representation allows observing the complete structure of the shop.

– **Interaction techniques:**

The interaction techniques added to the VGF allow the interaction with the data through an interface [20].

Interactive Zooming provides a scale diagram handling in two senses: nearing to the data to show information in detail on certain areas on the image; reduction of the image to give a wide perspective. In dependence of the original size the Gantt chart offers 25, 50, 75, 100, 150, 200 % of zooming.

Interactive Filtering consists on the selection of a data subset to observe its details, for example, to show the work-load assigned to each machine.

Linking and brushing provide simultaneous visualization of the same information in related diagrams that provide several perspectives of the same information to the user [25, 26].

4.3 Interface

The VGF interface integrates the visualization techniques and the interaction techniques mentioned previously.

The screen is divided in five areas (Fig. 4a): 1) Menus and shortcuts, 2) Gantt chart, 3) Shop tree, 4) Radar view, 5) Schedule properties.

The menus and shortcuts area incorporate basic actions: open files, save files, print diagrams, export a graphic schedule, as well as the shortcuts to provide greater agility in the handling of the system.

The tree diagram corresponds to the hierarchical structure of the shop, having the shop as root, and in second level the machines are located with the jobs as leaves. This diagram is related with the Gantt chart, the radar area and the schedule properties in the following way. When the user selects a node of the tree, it is shadowed in the Gantt chart and in the diagram of radar view. Also the interface shows the information details of the selected node according to the context of the element. For example, if the user selects the root of the tree, the Gantt chart stand out and the characteristics of the shop are shown in the area of properties containing: kind of shop (simple flowshop, hybrid or flexible flowshop), characteristics of machines (parallel, identical, uniform or not related machines), among other data (see Fig. 4a). If a machine is specified, then in the area of properties the characteristics of the suitable machine are shown containing information such as: number of jobs assigned for processing in a machine, completion time of all the works, jobs for stage with their respective completion time (see Fig. 4b). In the event of selecting a specific job, in the properties area are shown the instants of their release times in the shop, beginning and end of the setup times, and completion times (see Fig. 4c).

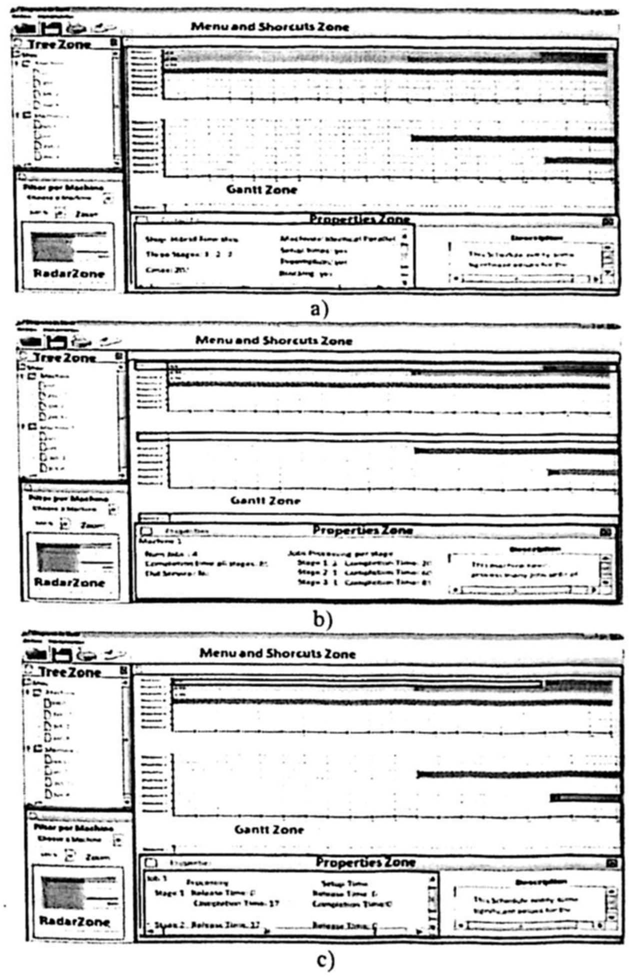

Fig. 4. VGF interface: a) selection of root tree; b) selection of machine; c) selection of job.

The Gantt chart spreads the schedule with all their characteristics defined in the input file of VGF: release time and finish time for each job, and their operations with regard to the machines. In this space it is possible to add the analyst's annotations about the graphic schedule.

Radar area shows the complete Gantt chart, adapted to the size of the area, using a reduced scale. It offers a wide perspective of the diagram, and allows the navigations to specified parts of the graphic representation in the area of Gantt chart.

In the area of schedule properties, some properties are shown according to the level selected in the area of shop tree. It also allows to the analyst to make comments about the schedule in description field.

All the diagrams that appear in the VGF interface are related to each other to offer different views from the same information, as it prescribes the use of the multiple visualizations, in such a way to provide bigger detail of data [27, 28].

4.4 Input Format

The computational system PLARETF provides to the VGF system with scheduled data in a special input format. The solution of the scheduling of a workshop, provided by PLARETF is not agile for understanding due to the extension of involved information. The input format of VGF was designed in such a way, that can adapt it to any scheduling context.

Below the input format implemented in VGF is described. The data is divided in four sections (Fig. 5).

The first section (Fig. 5a) contains the schedule notations. It is a summary of declarations of the notations in the text file for their visual representation.

The second section (Fig. 5b) summarizes the shop characteristics, which contains Graham's triplet: shop characteristics (FS, FFS or HFS), and characteristics of machines (parallel, identical, standardize, not related). Also, all the jobs characteristic and optimization criteria are specified.

The third section (Fig. 5c) describes the shop resources, detailing the number of stages and the number of machines for each stage, as well as the total jobs quantity in the shop. All the machines, jobs characteristics and optimization criteria are specified.

The fourth section (Fig. 5d) is the area of specification for schedule times. An array contains the timing of the works. The items correspond to concepts related with each job: processing time, release time, setup time, WIP, and blocking.

The columns are organized by triads. For an Item 1, the first triad describes the timing of job in the machine that carries out the first stage (operation): the machine number, time of beginning in this machine and time of the end. The second triad in the same way describes timing of the second stage, etc., for all stages and for all items.

The form of representation of input data in an array gives the flexibility to VGF because it is independent of the scheduling model.

The system allows to add more elements than they are in a traditional Gantt chart, that represent another data besides of jobs processing time and sequence depends setup times: in order to accomplish this, it will be enough with adding another label called description in the section one, and add an array in the section four for the new description.

4.5 VGF Architecture

The architecture of VGF is organized in three layers (see Fig. 6). The first layer corresponds to the interface by means of which the user interacts with the information, represents visually the algorithms results and allows to manipulate and explore the data. The second layer works as VGF administrator that monitoring and responds to the user's actions. The third layer contains the VGF component, divided in three modules: interpretation engine, entity data and mapping visual.

```
*****************************************************************
*                           NOTATIONS
*****************************************************************
      Fm        Taller de flujo
      FFc       Taller de Flujo Flexible
      FHc       Taller de Flujo Hibrido
      Pm        Maquinas Identicas
      Qm        Maquinas Uniforme
      Rm        Maquinas No relacionas
      st        Setup time
      prmp      Preemption
      prec      Precedencia
      brkdwn    Breakdown
      wj        priority
      Mj        Machine eligibility constraints
      prmu      Permutation
      block     Block
      nwt       no wait
      Cmax      makespan
      Ej        Earliness
      Lj        Lateness
      dc        Description

Note: An input value must put before the word " # " for the
      visualization could read and representing in display
```

a)

```
*****************************************************************
*                   Features of Calendar
*
*****************************************************************
          #FFc
          #Pm
          #st
          #prmp
          #block
          #Cmax

Note: In this space must describe all the general features of shop
```

b)

```
*****************************************************************
*                  Resources Specifications
*****************************************************************
#machines 8
#stage1 3
#stage2 2
#stage3 3

*****************************************************************
*                     Quantity of Jobs
*****************************************************************

#njobs 20

Note: In this space must be describe all the resources of shop,
      like machines, machines per stage, quantity of jobs.
```

c)

```
          p1  rj1   cj1     p2  rj2  cj2    ...    pn  rj   cj

Item 1    2   45    86      4   86   107    ...    5  107  115

Item 2    3  105   163      2  163   165    ...    6  165  179

Item 3    3   38    51      3   51    52    ...    6   52  125
  .                                          .
  .                                          .
Item n    2  128   161      7  161   169    ...    3  169  174
     Itemi= could be jobs, setup times
     ri= release time of item in processor or machine i
     ci= completion timeof item  in processor or machine i
     pi= processor or machine i

     i=0,n                      j=0,n
```

d)

Fig. 5. The VGF input format: a) notations section; b) schedule features; c) resources specification; d) array of processing times.

The interpretation engine is the one in charge of reading all the information of the input text file, to incorporate the information into the system and to store it in the data entity module. The mapping visual module take the stored information from data entity and transforms the information from scalar to the vector, and generate a diagram that is sent to the VGF interface to present it to the user.

The VGF component represents the Gantt chart. Therefore, it has the property of been reusable in other projects. It requires as input, the schedule information generated by the jobs scheduling algorithm and creates a Gantt chart with all the schedule's characteristics. The VGF component is independent of the management and visualization modules. It is designed to be re-used in any visual platform customizing its characteristics to respond to the necessities of future research where the visualization of job shops scheduling should be required.

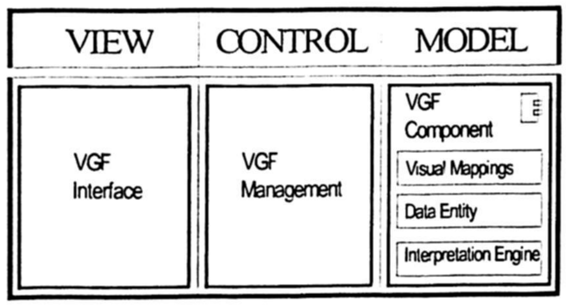

Fig. 6. VGF Architecture

5 Conclusion

In the development of visualization systems is important to take in account the specific context of the area in order to extract most information about the graphical representation of the data.

The real production implies necessities in the development and researching of scheduling algorithms with specific characteristic of the problem. The complexity in the visualization of calendars is due to the variety of elements to be represented in the same chart. At the moment doesn't exist software of free distribution that shows all the elements in a schedule. The commercial visualizers don't achieve with the investigation necessities and they have a high cost.

In this article we proposed a model of acquisition of the knowledge in the scheduling context through of the user's interaction with VGF. The design of the interface in form of four related areas gives to the user different perspectives of the information. VGF is a flexible visualization system due to his input data format, and visualizes diverse scheduling shops. This VGF could be reused in future researches that require the visualization of schedules generated for different scheduling models.

References

1. Johnson, S.M.: Optimal two- and three- stage production schedules with setup times included. Naval Research Logistics Quarterly 1(1), 61-68 (1954).
2. Garey, M.R., Johnson, D.S., Sethi, R.: The complexity of flowshop and jobshop scheduling. Mathematics of Operations Research 1(2), 117-129 (1976).
3. McKay, K. N., Pinedo, M., Webster, S.: Practice-focused research issues for scheduling systems. Production and Operations Management 11(2), 249-258 (2002).
4. Microsoft. Información General de Visio 2003.
5. http://www.microsoft.com/latam/office/visio/prodinfo/overview.mspx (October, 2007).

6. Mindjet. Mindget MindManager7.
7. http://www.mindjet.com/us/products/index.php (October, 2007).
8. Microsoft. Información General del Producto Microsoft Office Project Standard 2007 http://office.microsoft.com/es-es/project/HA101656383082.aspx (October, 2007)
9. Oracle. Oracle E-Business Suite. http://www.oracle.com/applications/e-business-suite.html (October, 2007)
10. Aqua eSolutions. Aqua EBS 2007. http://www.aquaesolutions.com/index.htm (October, 2007)
11. Deister. DEISTER ERP SUITE: Axional ERP. http://www.deister.es/es/products/e-erp/ (October, 2007)
12. Visual Consulting. Módulos Base de Visual Manufacturing.
13. http://www.visualconsulting.com.mx/visualmanufacturing.htm (October, 2007)
14. Ruiz, R., Şerifoğlu, F.S., Urlings, T.: Modeling realistic hybrid flexible flowshop scheduling problems. Computers & Operations Research 35(4) 1138-1150 (2008).
15. Pinedo, M., Chao, X., Leung, J. Michael Pinedo web site.
16. http://www.stern.nyu.edu/om/software/lekin/index.htm (March, 2008)
17. Romero, R.: Sistema computacional para la evaluación de algoritmos de planificación de trabajos en un taller de flujo híbrido. M.C. thesis, Engineering Institute, UABC, Mexicali, México (2007).
18. Yaurima, V., Burtseva, L., Tchernykh, A.: Hybrid Flowshop with Unrelated Machines, Sequence Dependent Setup Time and Availability Constraints: An Enhanced Crossover Operator for a Genetic Algorithm. In: Wyrzykowski et al. (Eds.) Parallel Processing and Applied Mathematics. LNCS, vol. 4967, pp. 609-617. Springer (2008).
19. Leung, J. Y-T.: Handbook of scheduling: algorithms, models and performance analysis. Chapman and Hall/CRC, Fl. (2004).
20. Pinedo, M.: Scheduling: theory algorithms and systems. Prentice-Hall, New Jersey (2002).
21. Dursteler, J. C.: Visualización de información. Gestión 2000, España (2001).
22. Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy R.: Advances in Knowledge Discovery and Data Mining. American Association for Artificial Intelligence, LA (1996).
23. Wurman, S. R.: Information Architects. Watson-Guptill Pubns, NY (1997).
24. Colin, W.: Information Visualization: Perception for design. Morgan Kauffman, San Francisco (1999).
25. Card, S. K., Mackinlay, J. D., Shneiderman, B.: Readings in information Visualisation: Using Vision to Think. Morgan Kaufmann, San Francisco (1999).
26. Keim, A. D.: Information Visualization and Visual Data Mining. IEEE Transactions on Visualization and computer Graphics 7(1), 100-107 (2002).
27. Keim, A. D.: Visual Techniques for Exploring Databases. Int. Conference on Knowwledge Discovery in Databases. http://www.dbs.informatik.uni-muenchen.de/~daniel/publication.html (1997)
28. Wainer H., Velleman, P.: Statistical graphics: Mapping the pathways of science. Annual Review of Psychology, 52, 305-335 (2001).
29. Swayne, D., Lang, D.T., Buja, A., Cook, D.: GGOBI: Evolving from XGobi into an extensible framework for interactive data visualization. Computational Statistics & Data Analysis, 43(4), 423-444 (2003).
30. Young, F.W.: ViSta: The Visual Statistics System. http://www.visualstats.org (March, 2008)
31. Ledesma, R., Molina, J. G., Young, F. W., Valero, P. M.: Desarrollo de Técnicas de visualización Múltiple en el programa Vista: Ejemplo de aplicación al análisis de componentes principales. Psicothema 19(3), 497-505 (2007).
32. Young, F. W., Faldowski, R. A., McFarlane, M. M.: Multivariate statistical visualization in computational statistics. In C.R. Rao (ed.): Handbook of Statistics, Amsterdam: Elsevier Science, pp. 959-998 (1993).