Deep Drawing Tool for E-learning

A didactic approach for manufacturing engineering education

Francisco Javier Ramírez
Faculty of Computer Science Engineering
Universidad de Castilla-La Mancha
Albacete, España
Franciscoj.ramirez@uclm.es

Rosario Domingo and Miguel Angel Sebastián
Dept. of Manufacturing Engineering
UNED
Madrid, España
rdomingo@ind.uned.es; msebastian@ind.uned.es

***Abstract*— Manufacturing processes are an important element in industrial engineering education. In distance education, the learning of engineering subjects has a special difficulty, which can be reduced by means of the use of new technologies, and the practice of mixed models of learning. One of these processes is the deep drawing due to its relevance in the industry. This paper presents a deep drawing tool for e-learning. The tool has been realized for its use in the Master degree because it requires advanced knowledge in manufacturing processes. The instrument has been developed with the objective of the students who can: a) Select input data for get the formability of material to deep drawing; b) Select the process that provides the best solution from a technological perspective; c) Optimize the process for saving the material; d) Know the influence of the punch in the results; e) Consideration of the process cost. The structure of the system has three subsystems: a) Solve, module for data processing and the generation of results; b) Materials, module for management data of the system; and c) Interface, module for user interaction. The tool has been implemented in the software tool programming, developed in Java. This language has been selected because it provides a methodology of object-oriented programming and its execution is possible in multiple operating systems. The paper describes each step of the tool, from the input data to final analysis and they are shown through the results given by the tool.**

Keywords-component; engineering education; manufacturing; manufacturing data processing; manufacturing planning; computer aided engineering formatting.

I. INTRODUCTION

Manufacturing processes are an important element in industrial engineering education. In distance education, the learning of engineering subjects has a special difficulty, which can be reduced by means of the use of new technologies, and the practice of mixed models of learning [1]. One of these processes is the deep drawing due to its relevance in the industry [2]. This paper presents the development of a didactic computer tool created by the authors for the resolution of multi-stage deep drawing industrial processes. Thus, the tool has been realized for its use in the Master degree because it requires advanced knowledge in manufacturing processes. Although there are some software applications about the deep-drawing optimization, this one allows analyzing technological and economical constrains and it permits minimizing the total process time, from a global perspective. The model provides a comprehensive analysis occurring in multi-stage processes of axi-symmetric geometry work-pieces [3-5]. The scientific model is established from the Leu [6] and Sonis et al. [7] works that provides LDR (limiting drawing ratio) solutions based on normal anisotropy value, strain hardening exponent and others, applied to the drawing and redrawing stages. The authors extend this work to the ironing stages, and provide a global and integral science solution for the total process [8-9]. The model permits to modify and correct certain process variables in order to predict the impact of those that are not fully controllable. It is based on defining a set of boundary conditions of the process for determining a range within which it is stable. The simultaneous accomplishment of the boundary conditions involved in each process (deep drawing, redrawing and ironing) permits to limit a range of values that define each stage of the total process [10-11] (see Fig. 1).

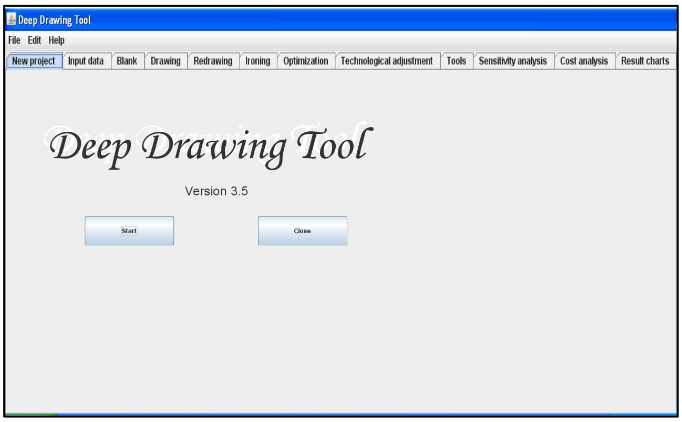


Figure 1. Main screen of the computer tool

Based on the technological model formulated, this paper presents a software tool created to implement the Aided System in a practical way. The computer tool was generated in a Java environment by creating a user-friendly interface and easy in its operation. The methodology has been the spiral model, software development model of evolution in which the software is realized in a series of incremental versions. This paper describes the tool constructed in their requirements, development methodology and structure. Finally, applies to the resolution of a case study, with the resolution screens that provides the Assisted System.

II. System Requirements

The system must comply several needs and characteristics of high-level of the developed system. The system must:

- Be independent of particular operating system. This should run on any system available in the market.
- Show the calculation results in a reduced time allowing easy interaction between user and system.
- Have a simple, clear and easy operation.

The interface design must meet the following general principles: a) Ease of learning: the user does not need much training time to use the interface; b) Consistency: all mechanisms of the interface are always used in the same way and thus produce the same result; and c) Flexibility: allow user interaction to drive the system through messages indicating everything that happens during the interaction.

The application consists of three parts:

- Calculation Module for data processing and generation of results.
- Materials: Module for the management of various materials that the system works.
- Interface Module intended for user interaction.

III. Methodology

The spiral model is a model of evolutionary software development where software is developed in a series of incremental versions. In the initial approach the incremental release might be a paper model or prototype as they are already performing approaches are achieved in more complete versions of the system. This model is suited to create products with different versions, improving with each approach, the current version and adding new functionality. Engineering can be developed through classical life cycle model or prototype construction. This methodology will, in each cycle, review the specifications of objects based on knowledge that is acquired with the operation of the aided system.

The spiral model is divided into regions of structural activities or tasks. There are generally three to six areas of work. These task areas are listed below:

- Customer communication: It includes the tasks necessary to obtain customer requirements. In this case a customer does not exist as such; the requirements are obtained from the project definition.
- Planning: Contains the tasks required to prepare the project, define resources and estimate the development time.
- Risk analysis: It includes the tasks required to assess technical risk and project-related tasks.
- Engineering: Contains the tasks required to set the system to be developed.
- Construction and adaptation: It contains the tasks required to build, test, install and provide user support.
- Customer Review: It includes the tasks required for the degree of customer satisfaction with the developed system.

When the evolutionary process starts, the software engineering team revolves around the spiral in clockwise direction, starting from the center. The first circuit of the spiral causes the development of a product specification; the following steps on the spiral could be used to develop a prototype and progressively more sophisticated versions of the software. Each step of the planning region produces adjustments in the project plan. Accordingly, the cost and the planning are adjusted by the reaction to the evolution of the client.

The spiral model can be adapted and applied throughout the life of the software. As the software evolves and the process progresses, the developer and the client understand and react to risks at each evolutionary level. This model allows the developers to apply the prototyping approach at any stage of product evolution. It will consider technical risks at all stages of development and reduce them before they compromise the project's development.

IV. Structure of the System

The purpose of this section is to describe the structure of the system, which has three packages that define it.

Each one of these packages corresponds to a module defined in the system.

- Calculation Package: Package that implements the functions of data processing and delivery of outcomes.
- Material Package: Package that implements the management functions of the several materials that the system works.
- GUI Package: Package that implements the user interface.

Each of these packages in turn contains classes that implement the functionality itself.

V. Development of the Computer Tool

The purpose of this section is to define the precise nature of the development system used.

Since the system must be compatible with different operating systems in the market, we have selected Java as development language. Java is a language of object-oriented programming developed by Sun Microsystems in the early 90s. The Java language was created with five main objectives:

- It should use the methodology of object-oriented programming.

- It should allow the execution of a program on multiple operating systems.
- It should include default support for networking.
- It should be designed to execute code on remote systems securely.
- It should be easy to use and take the best from other object oriented languages like C + +.

Note that Sun Microsystems released the bulk of their Java technologies under the GNU GPL, according to the specifications of the Java Community Process, so that virtually all of Sun Java is now free software. To facilitate the development and purifying, the system will use a modular development system.

Modular programming can be defined as one that addresses the solution of a problem by breaking it down into simpler sub-problems, each of which is solved using an algorithm or module more or less independent of the rest (hence the name "modular programming"). The advantages of modular programming are several: facilitates the understanding and resolution step; increases the clarity and readability of programs; allows multiple programmers working on the same problem at a time, since each one can work in one or more modules quite independently; reduce development time, reusing previously developed modules; improving the reliability of programs, it is easier to design and debug small modules than huge programs; and finally facilitates the software maintenance.

Moreover, we can safely say that it is virtually impossible to write a large program if we do not proceed to divide it into smaller chunks, covered by our poor human intellect. In general, the main problem is solved on an algorithm which we call algorithm or core module, while the simple sub problems are solved in sub-algorithms, also called modules. Sub-algorithms are subordinate to the main algorithm, so it is what decides in which order it should be implemented and how the sub-algorithms dataset. The main algorithm makes calls or invocations to sub-algorithms, while these return results to the former. Thus, the main algorithm is collecting all the results and it can generate the solution to the global problem.

VI. Application to a Case Study

This section shows the resolution of a case study developed by the Aided System. As input data following the dimensions of the piece that we want to obtain:

- External diameter $d_n = 150$ mm
- Length $l_n = 600$ mm
- Bottom thickness $s_n = 4$ mm
- Wall thickness $e_n = 1$ mm
- Material type UNS A95182.

A. *New Project*

Once implemented the program shows the startup screen. We will begin to work with the tool clicking on the START button (Fig. 1). If we click CLOSE the application will be closed. This screen also indicates the application title "Deep Drawing Tool" and the version we're using, in this case 3.5. From the home screen we can access at any stage after having run through the dialogs above, showing the titles of the corresponding phases.

B. *Input Data*

The input data are entered on screen "INPUT DATA" (Fig. 2), which indicates the size and type of piece we want to get. The dialog asks us about several data concerning the external diameter, length, bottom and wall thickness.

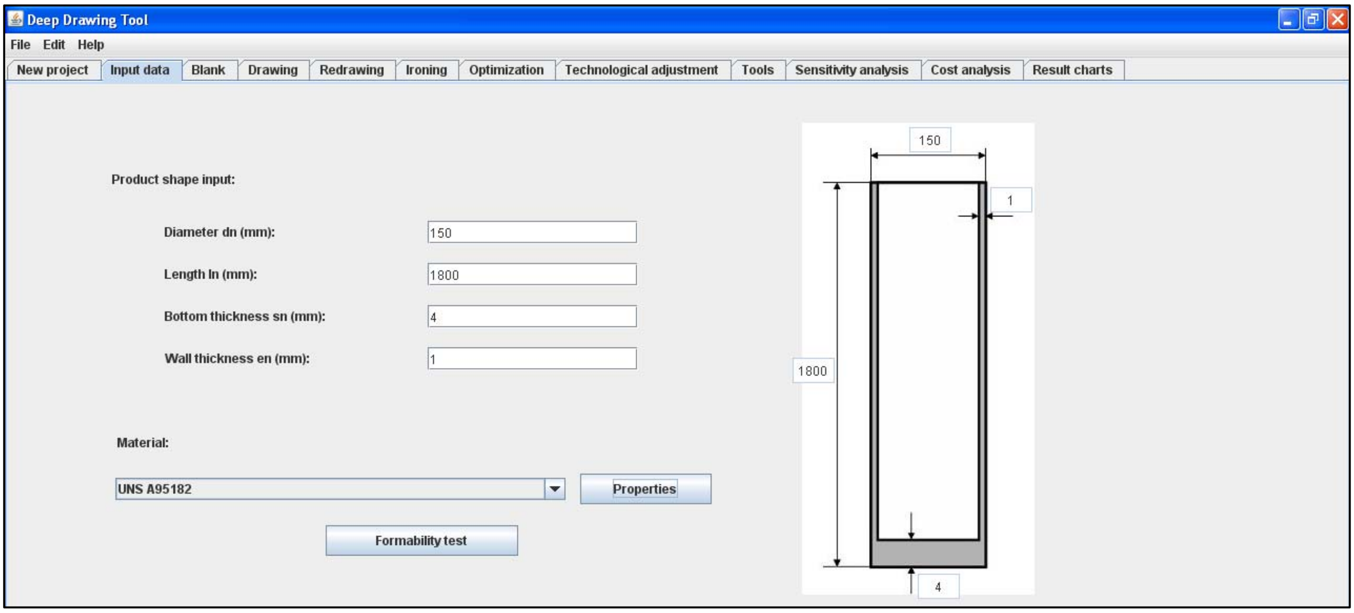


Figure 2. Input Data Screen

We must also choose the type of material in which we want to make the piece, with a choice between several options. In this version of the program we have selected three materials: UNS C26000 brass, UNS G10080 steel and UNS A95182 aluminum. For the present case we have selected UNS A95182 aluminum. Clicking on the PROPERTIES button it will be open a window informing us about the characteristics of the selected material (Fig. 3).

C. Blank Dimensions

The second step of the application is the study of the blank dimensions. The application displays the screen for determining the dimensions of the disc or BLANK (Fig. 4). For information purposes, this screen shows the data corresponding to the final dimensions the piece will have and the type of material chosen.

Figure 3. Material Properties Screen

By pressing the button "SOLVE" we can obtain the dimensions of the blank: diameter and thickness.

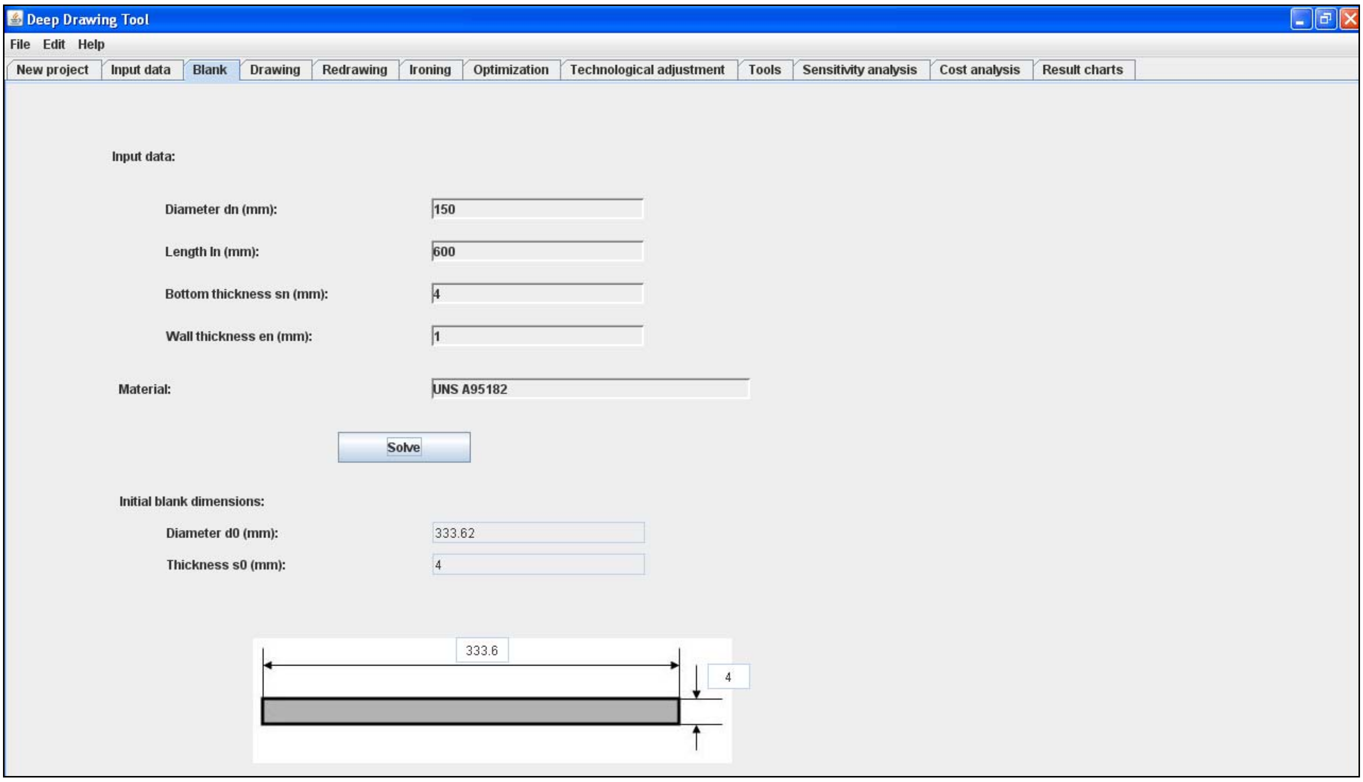


Figure 4. Blank Dimensions Screen

D. Drawing

The size of the disk obtained starting the program solves the first phase of the process of drawing or DEEP DRAWING, through the screen shown in Fig. 5.

The input data are those relating to blank diameter, thickness and diameter of the punch-stretching phase. It is important to show here the value of this diameter d_p because it will condition the design of this phase. It may be that the boundary conditions at this stage of drawing are to obtain a diameter smaller than the diameter of the punch inside the piece that we want to manufacture, an issue that limits the calculation at this stage [12].

The application requires the introduction of the value for efficiency coefficients in the two boundary conditions defined. The remaining input values are shown for information, since the application's taken directly from the database, depending on the type of material selected. SOLVE pressing the program calculates the drawing stage called Stage 1.1, providing the necessary data regarding the diameter of the stuffing, length, thickness, diameter of the punch, drawing ratios and thickness reduction, and maximum force applied during the process drawing.

Once it defines the drawing phase, the program performs the audit of the experimental conditions of drawing, for information.

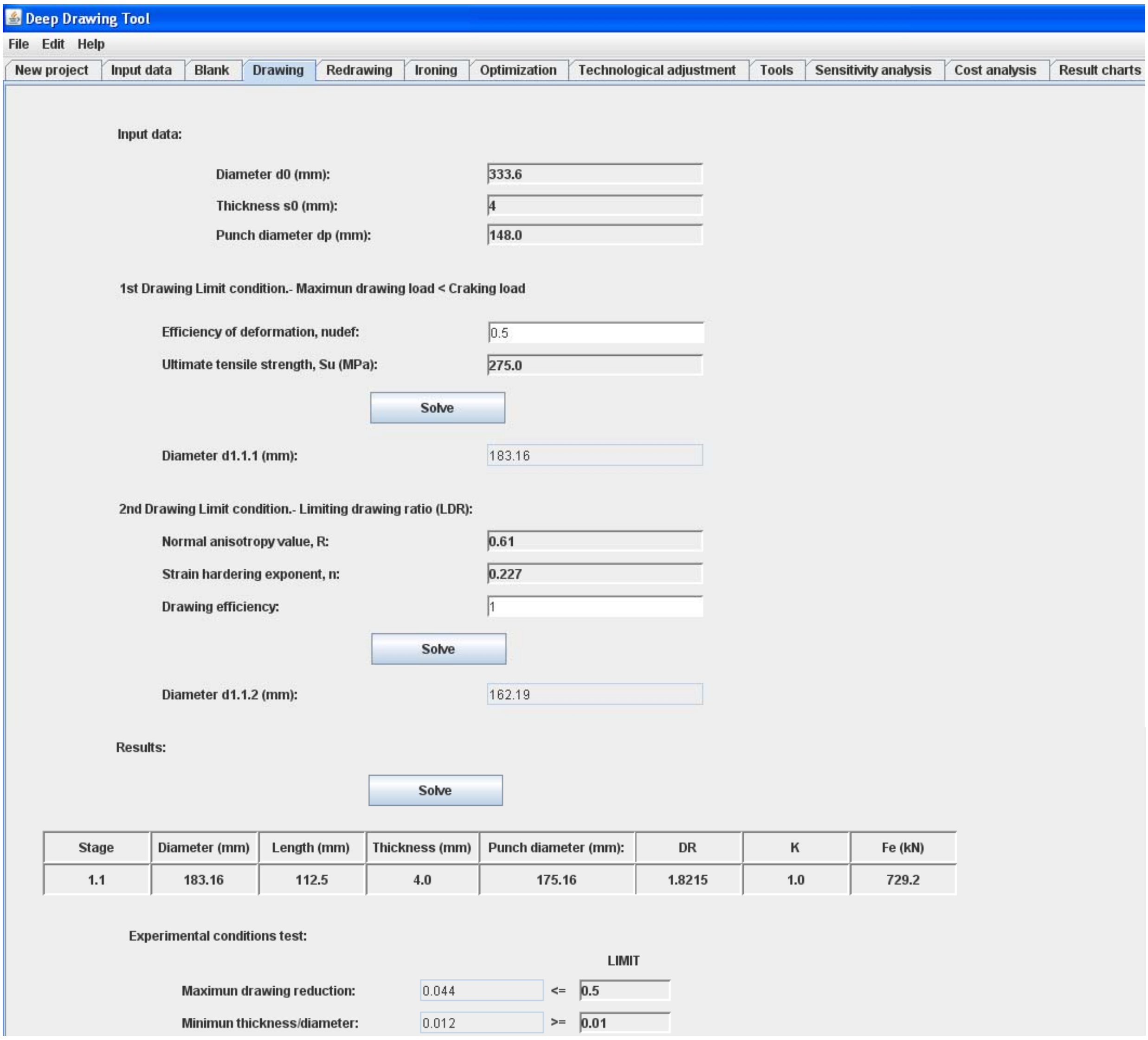


Figure 5. Drawing screen

E. *Redrawing*

The next phase of the process resolves redrawing or redrawing operations. From the dimensions obtained in the previous phase, embossing, which were provided as input, the program models at this stage on three boundary conditions (Fig. 6). In the first condition it is necessary to introduce limit values for the coefficient of friction, angle of the matrix and the radius die. The second boundary condition requires the application to provide the data for the efficiency factor and the value of the initial deformation [13]. The third condition is resolved based on the information that counts and that the application has already been introduced earlier. SOLVE proceed by pressing the calculation of phases for the process redrawing from 1.2 to Stage 1.n. It may be that it only requires a redrawing phase, as in the case presented, or even none.

In the event that, by the geometry of the piece to model, we evaluated in the case that it is not necessary redrawing stages, the program will provide a message that this case does not require redrawing phases.

The resolution redrawing process provides the results for the diameter of the stages, length, thickness, diameter of the punch, redrawing ratios, maximum force exerted at each stage and radio redrawing entry in the matrix for each phase.

The application also provides the results for Phase 1.m is that stage at which we could have come with redrawing process if the final dimensions of the piece had not limited the process. Redrawing, when the process is conducted in three stages over this "surplus ratio" in redrawing process is used to optimize the corresponding phases.

F. *Ironing*

The next phase of the model for the drawing process is solved by the application of three drawing boundary conditions (Fig. 7). The first boundary condition is solved in a straightforward way by applying the data for the material and geometry of the last stage of the process of redrawing. In the second boundary condition requires the application to provide the data for the efficiency factor and the value of the initial deformation. The third condition corresponding to the maximum ratio of wall thickness reduction is resolved based on the information that counts and that the application has already been introduced earlier. SOLVE proceed by pressing the calculation of phases for the drawing process, from Stage 2 to Stage n. It may be that it only requires a drawing phase, or even none will be necessary.

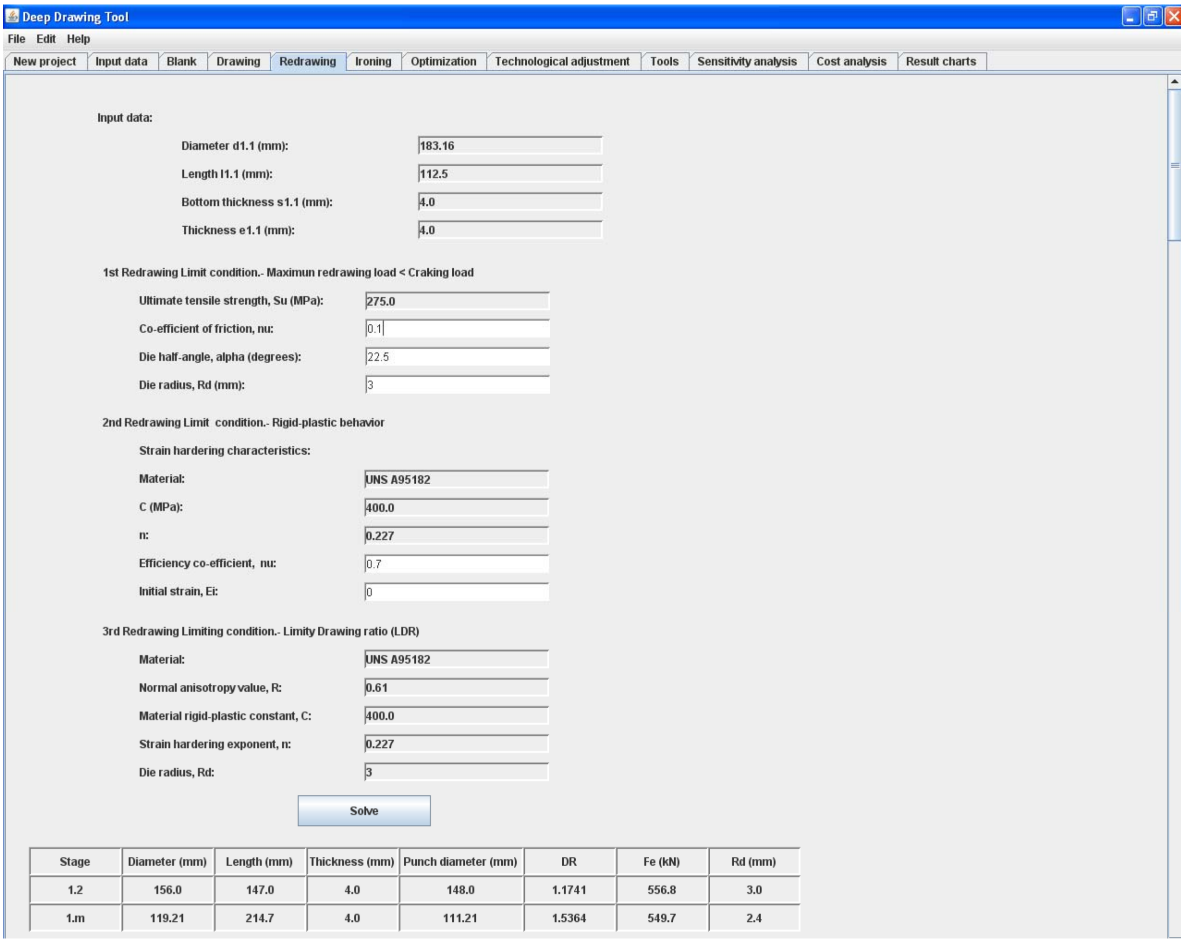


Figure 6. Redrawing screen

G. *Process Optimization*

Once provided the initial solution, the model is an optimization process in a comprehensive manner for redrawing and ironing operations (Fig. 8).

As input ratios appear stretched redrawing and surpluses in the last stage of each process. The model solves the optimization based on the definition of the number of times that it is necessary to split the surplus ratio (x_{re} for redrawing and x_e process for the drawing process), and operating parameters of the machinery involved in every stage of process [14-16].

Before proceeding to calculate the optimization process it is necessary to set the operating parameters of the machinery, for which the system will ask for entering the machinery used in each stage:

- Approach velocity
- Operation velocity
- Recovery velocity
- Tool total length

These data are entered via the dialog screen shown in Fig. 8. After entering the operating parameters of the machine, pressing SOLVE it proceed to the calculation of the optimization process, the following output shown in Fig. 9.

The results are the outcomes for each phase, optimized processes and redrawing stretched, diameter, length, thickness, drawing ratio, ratio of wall thickness reduction, drawing maximum power and operating time.

H. *Technological Adjustment*

The model allows adjustment technology by defining a clearance that increases the diameter of the hallmarks of the stages of drawing. Defined this clearance, the model automatically recalculates the entire process and provides a solution "adjusted" for the entire process: drawing, redrawing and ironing (Fig. 10).

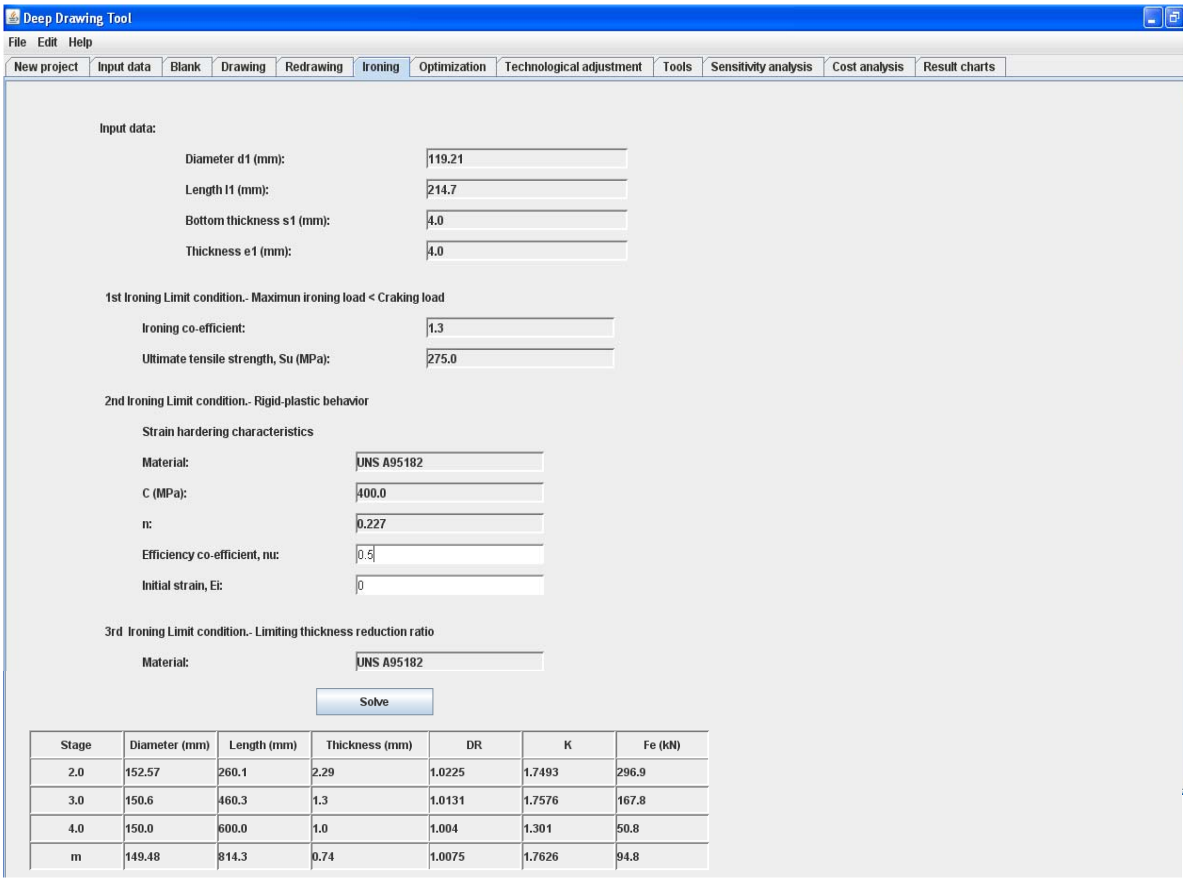


Figure 7. Ironing screen

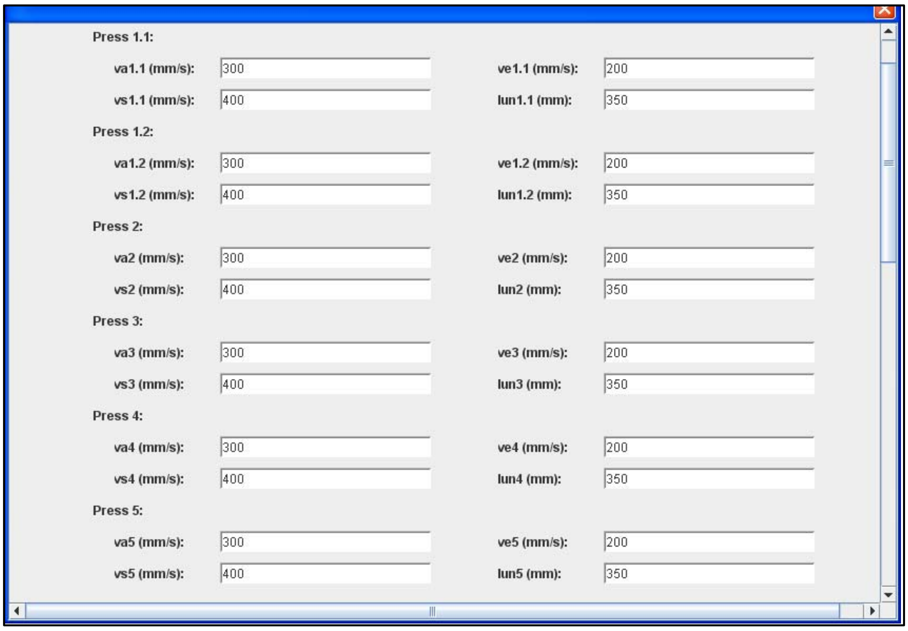
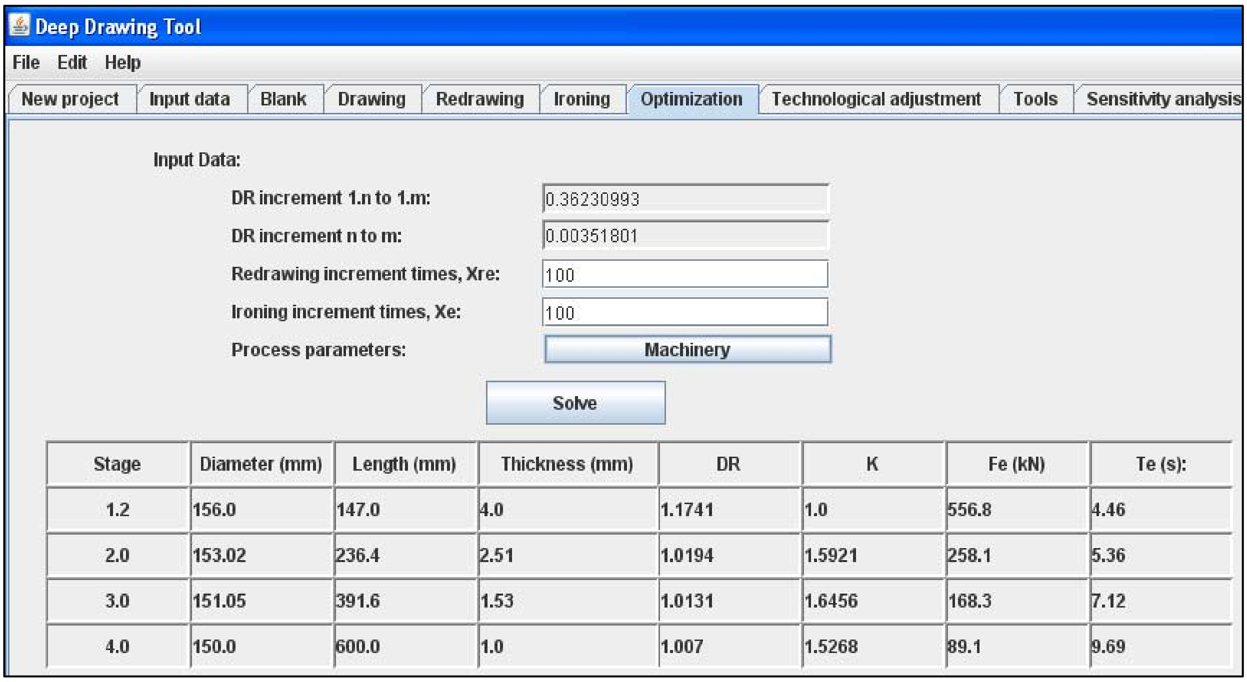


Figure 8. Machinery parameters

I. *Sensitivity Analysis*

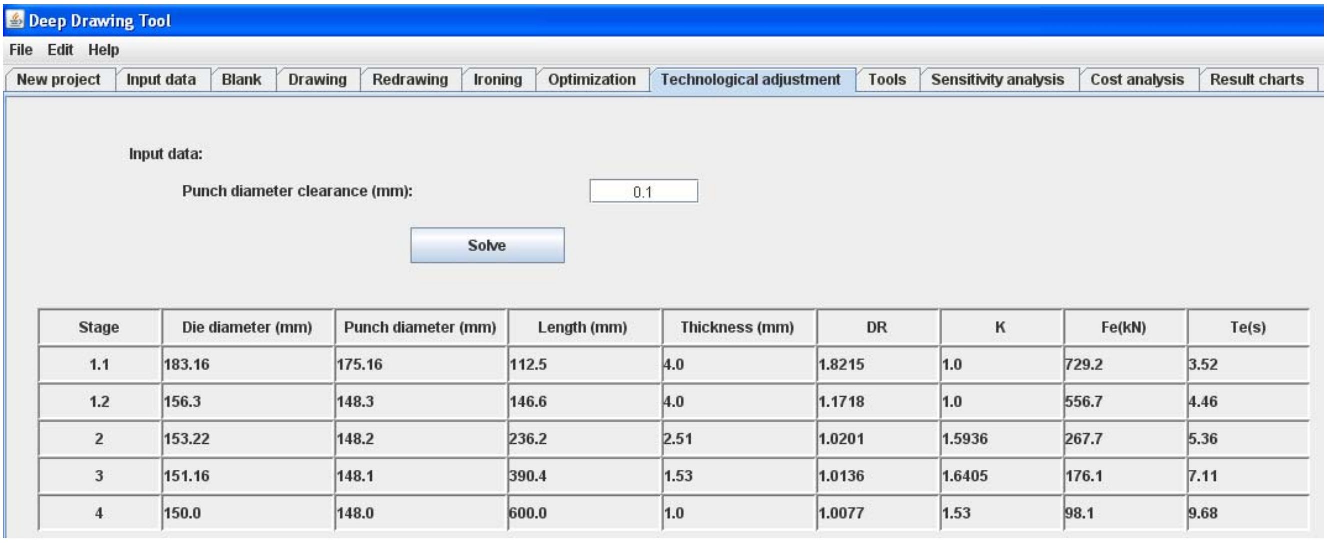
In the next phase, the process carries out a sensitivity analysis to the weathering of parent involvement in the process.

We define the diameter limit for each stage which can progress to the wear of the dies without endangering the stability of the process. Having established the limiting diameter setting a check is made as the diameter of the stage corresponds to the limit calculated (Fig. 11).



Stage	Diameter (mm)	Length (mm)	Thickness (mm)	DR	K	Fe (kN)	Te (s):
1.2	156.0	147.0	4.0	1.1741	1.0	556.8	4.46
2.0	153.02	236.4	2.51	1.0194	1.5921	258.1	5.36
3.0	151.05	391.6	1.53	1.0131	1.6456	168.3	7.12
4.0	150.0	600.0	1.0	1.007	1.5268	89.1	9.69

Figure 9. Optimization Process Screen



Stage	Die diameter (mm)	Punch diameter (mm)	Length (mm)	Thickness (mm)	DR	K	Fe(kN)	Te(s)
1.1	183.16	175.16	112.5	4.0	1.8215	1.0	729.2	3.52
1.2	156.3	148.3	146.6	4.0	1.1718	1.0	556.7	4.46
2	153.22	148.2	236.2	2.51	1.0201	1.5936	267.7	5.36
3	151.16	148.1	390.4	1.53	1.0136	1.6405	176.1	7.11
4	150.0	148.0	600.0	1.0	1.0077	1.53	98.1	9.68

Figure 10. Technological Adjustment Screen

J. Cost Analysis

In this phase of the system evaluated the costs involved in the whole process by introducing the input data for the total number of units to manufacture, raw material costs, hourly cost of labor and cost of electricity. The model provides the total costs for different solutions: initial, optimized and technological adjustment (Fig. 12).

K. Results

Finally, the model shows the graphs of results showing the evolution of certain variables over the entire process and showing a comparative for the solutions provided: initial, optimized and technological adjustment.

The graphics are as follows:

- Drawing Load vs Stage........................ Fig. 13
- Drawing Ratio vs Stage……………… Fig. 14
- K coefficient vs Stage……………… Fig. 15
- Length vs Stage……………………… Fig. 16
- Process total time…………………… Fig. 17

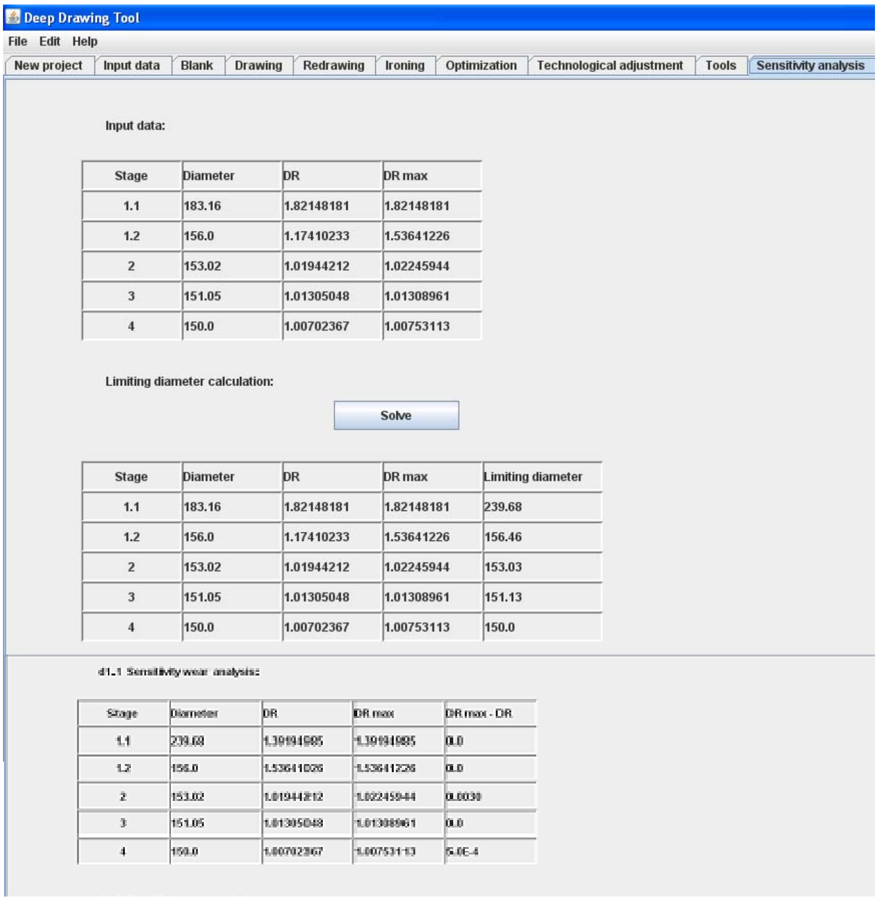


Figure 11. Sensitivity Analysis Screen

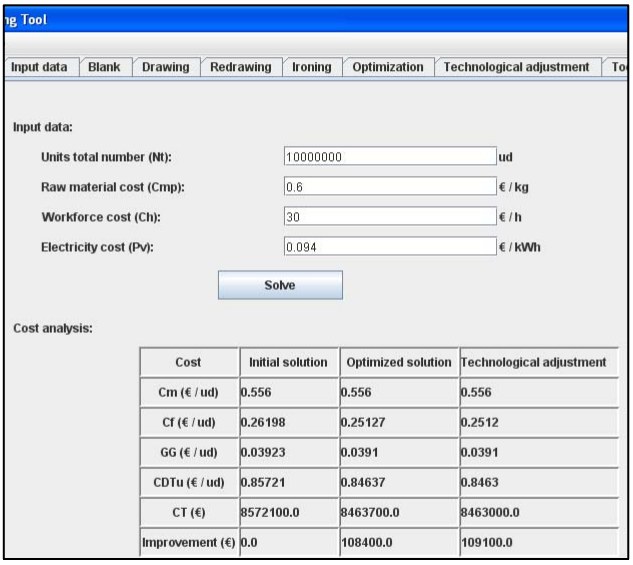


Figure 12. Costs Analysis Screen

CONCLUSIONS

This paper has summarized the functions of a deep drawing tool for engineering education. The tool allows selecting input data for getting the formability of material to deep drawing, selecting the process that provides the best solution from a technological perspective, optimizing the process for saving the material, knowing the influence of the punch in the results and considering the process cost.

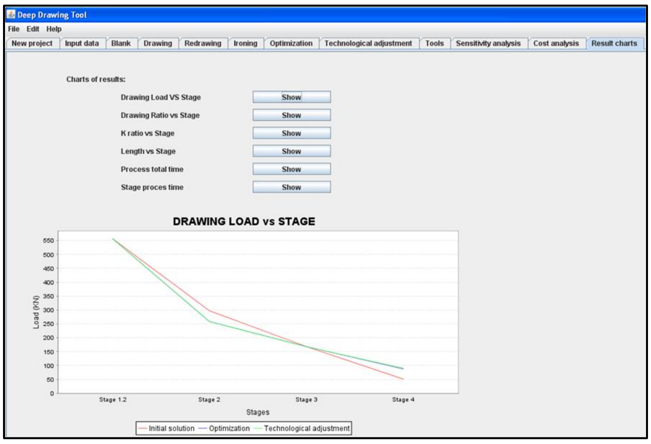


Figure 13. Graphic "Drawing Load vs Stage"

The structure of the system has three subsystems: Solve, module for data processing and the generation of results; Materials, module for management data of the system; and Interface, module for user interaction. The tool has been implemented in software tool programming, developed in Java. The paper describes each step of the tool, from the input data to final analysis and they are shown through the results given by the tool.

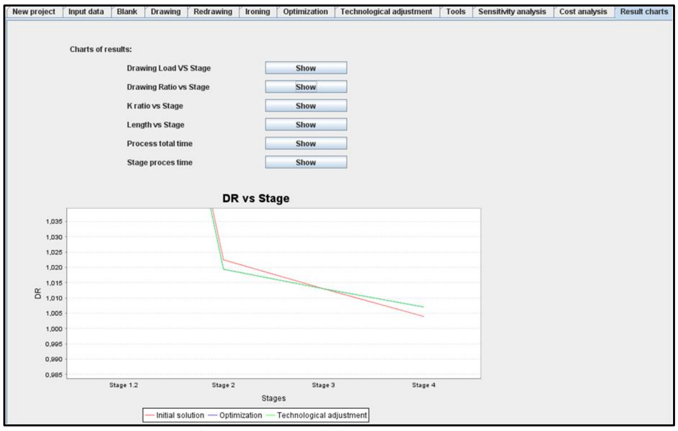


Figure 14. Graphic "DR vs Stage"

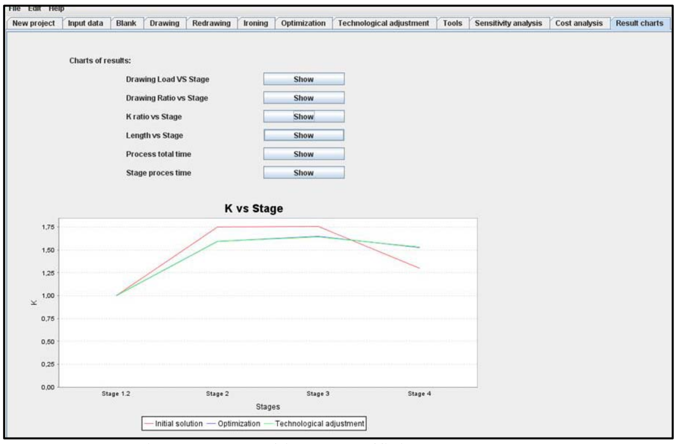


Figure 15. Graphic "K Coefficient vs Stage"

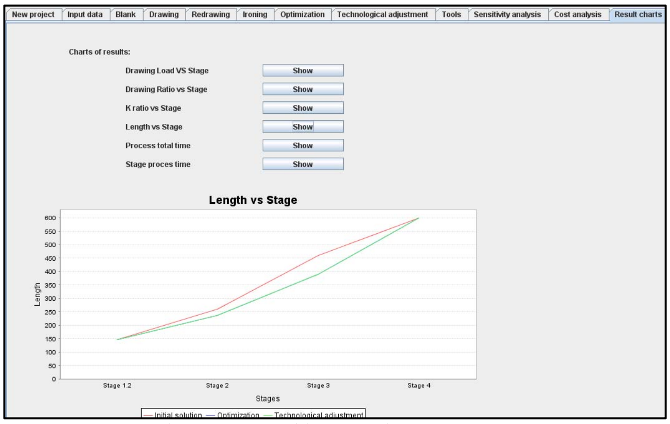


Figure 16. Graphic "Length vs Stage"

Future works, in this field, will go towards to implementation of the tool, and accordingly to the results, some improvements will be carried out.

REFERENCES

[1] M. Castro, E. L. Aldea, F. Garcia-Sevilla R. Gil, E. Sancristobal, G. Diaz, A. Vara, and J. Peire, "New laboratory working procedure considerations for electronic learning in a mixed (distance/online/traditional) model" 13th IEEE Mediterranean Electrotechnical Conference (MELECON) Spain, p. 1217, 2006.

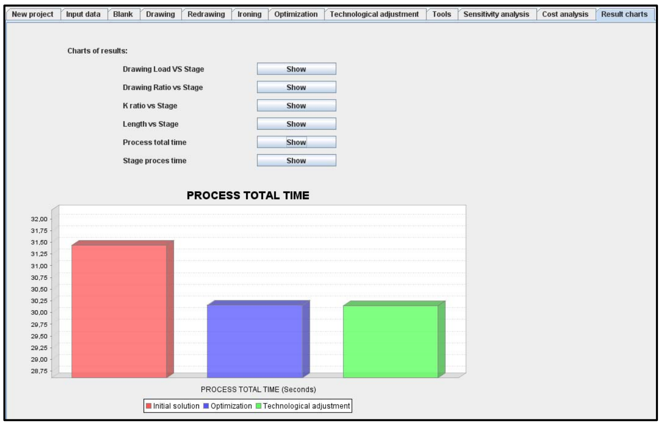


Figure 17. Graphic "Process total time"

[2] F. J. Ramírez, and R. Domingo, "Application of an aided system to multi-step deep drawing process in the brass pieces manufacturing" MESIC Spain, p. 8, 2009.

[3] K. Lange. Handbook of Metal Forming. McGraw-Hill, 1985.

[4] H. Ford. Advanced Mechanics of Materials. Ellis Horwood Limited, 1986.

[5] M.A. Hassan, N. Takatura and K. Yamaguchi, "A novel technique of friction aided deep drawing using a blank-holder divided into four segments", J. Mater. Proces. Technol., vol. 139, pp. 408-413, 2003.

[6] D.K. Leu; "Prediction of the limiting drawing ratio and the maximum drawing load in cup-drawing", Int. J. Mach. Tools Manuf., vol. 37, pp. 201-213, 1997.

[7] P. Sonis, N. Venkata N. and G.K. Lal; "On multistage deep drawing of axisymmetric components", J. Manuf. Sci. Eng., vol. 125, pp. 352-362, 2003

[8] A. Agrawal, N. Venkata and P.M. Dixit, "Determination of optimum process parameters for wrinkle free products in deep drawing process", J. Mater. Process. Technol., vol. 191, pp. 51-54, 2007.

[9] K. Rahul and C. Sanjay, "An improved model for predicting limiting drawing ratio", J. Mater. Process. Technol., vol. 172, pp. 218-224, 2005.

[10] T.H. Choi, S. Choi, K.H. Na, H.S. Bae and W.J. Chung, "Application of intelligent design support system for multi-step deep drawing process", J. Mater. Process. Technol., vol. 130-131, pp. 76-88, 2002.

[11] K. Hiroski, M. Kenichi and Y. Syoichiro, "A database oriented process control design algorithmm for improving deep-drawing performance", J. Mater. Process. Technol., vol. 138, pp. 343-348, 2003.

[12] S. Kalpakjian and S.R. Schmid. Manufacturing, Engineering and Technology. Prentice-Hall, 2007.

[13] M.A. Sebastián and A.M. Sánchez-Pérez. Diseño asistido por ordenador de los útiles para la embutición profunda de piezas cilíndricas huecas. E.T.S.I.I, UPM, Madrid, 1980.

[14] H. Sattari, R. Sedaghati and R. Ganesan, "Analysis and design optimization of deep drawing process. Part II: Optimization", J. Mater. Process. Technol., vol. 184, pp. 84-92, 2007.

[15] E. Siebel and H. Beisswänger. Deep Drawing. München, Carl Hanser, 1995.

[16] B. Endelt, K.B. Nielsen and J. Danckert, "New framework for on-line feedback control or a deep-drawing operation", J. Mater. Process. Technol., vol. 177, pp. 426-429, 20006.