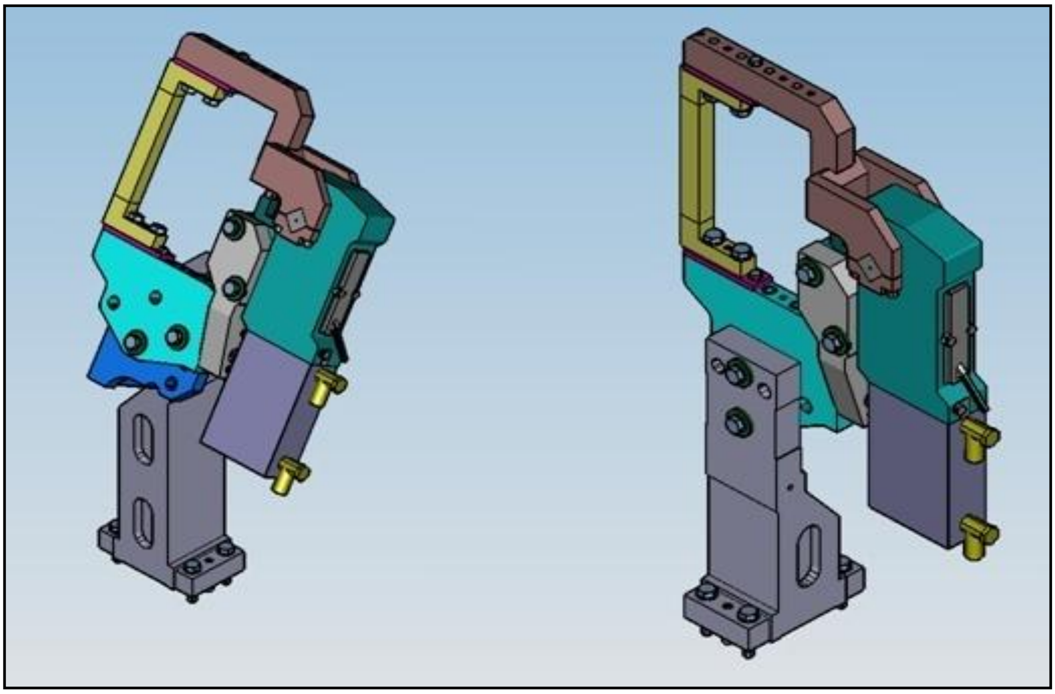
2008: ROB03

Modular Fixture Design for BIW Lines Using Process Simulate

Ali Keyvani



Master Thesis
Mechanical Engineering, Master of Robotics Program
Department of Technology, Mathematics and Computer Science

Modular Fixture Design for BIW Lines Using Process Simulate®

Summary

The unchangeable need of securing and locating parts during different manufacturing processes turned the fixtures to key elements in many part production industries. The iterations between design engineers and manufacturing planners because of late collision detection of the part/fixtures with robots cost a lot of time and money. The lead-time can be reduced by developing tools and/or methods for early verification of the fixtures during the simultaneous engineering phase. Different aspects of fixture designing, modeling and simulating is investigated as a base step to recognize the best practice work to do fixture planning in Process Simulate integrated PLM environment. The aim of the project is to use Process Simulate to design and validate modular fixtures at the same time and in a single environment. It also aims to investigate the possibility of adding kinematics, sensors, and actuating signals to the fixtures and utilize them to model the fixture behavior in a larger simulation study. The project narrows down its focus on the fixtures designed for robotic applications specifically in Automotive Body in White lines without losing generality.

The document type stated at the title page and in the header of this page is *master thesis work*.

Author:	Ali Keyvani		
Examiner:	Fredrik Danielsson		
Advisor:	Tobias Persson, SAAB		
Programme:	Master programme in Robotics		
Subject:	Thesis Project	**Level**:	Advanced, 15 ECTS
Date:	June 16, 2008	**Report Number**:	2008: ROB03
Keywords	Virtual Manufacturing, Modular Fixture, Automate Fixture Planning, PLM software, Collision Detection, Fixture Modelling, Process Simulate, Process and Manufacturing		
Publisher:	University West, Department of Technology, Mathematics and Computer Science, S-461 86 Trollhättan, SWEDEN Phone: + 46 520 22 30 00, Fax: + 46 520 22 32 99 Web: www.hv.se		

Preface

I would like to take this opportunity to acknowledge the people whose contributions and assistance helped tremendously in the completion of this research. I would like to show my appreciation to Dr. Fredrik Danielsson, my supervisor and advisor for his supports and guides. I would also like to thank Mr. Tobias Persson from SAAB for his arrangements and attentions.

I also thank the great help offered by Dan Svensson from SAAB and Hans Hörfelt from Volvo Cars, which provide me information about MODFix project.

Finally, I would also like to express my sincere thanks to all of my friends who have been helping me during the course.

Contents

List of Symbols

API: **a**pplication-**p**rogramming **i**nterface

BiW: **b**ody **i**n **w**hite

CAD: **c**omputer **a**id **d**esign

FMS: **f**lexible **m**anufacturing **s**ystems

MFG: **m**anufacturing **f**eatures

PLM: **p**roduct **l**ifecycle **m**anagement

PLP: **p**rinciple **l**ocation **p**oint

XML: e**x**tensible **m**arkup **l**anguage

1 Introduction

In most of manufacturing processes[1], the product should be fixed securely and accurately in front of the machine in order to desire task can be accomplished. This fixing duty is the main role of fixtures in the industry. Fixture designing is an inevitable common task of both design and process planning engineers and demands considerable work, time, energy, and money. Traditional disciplines suggest the order of part design, fixture design, and then going down to manufacture planning procedure; which means that in 'work holding problem', the part, and not the process is the primary consideration (1). However, this method is under serious debate recently due to variety of products and need for accurate process planning far in advance. This report will describe the need for opening a new horizon in this topic.

1.1 Flexible Manufacturing and Flexible Plants

Presence of robots as a resource to the manufacturing process has been tied with the term Flexible Manufacturing Systems (FMS). According to the definition (2) FMS is:

> "*A group of machining, assembly or general fabrication stations, each supplied with components by a variable mission conveyor system, and all controlled by computer*".

Or (3)

> *"The ability of a manufacturing system to adapt to changes in environmental conditions and in the process requirements"*.

With the aid of FMS, manufacturers are now capable of changing their product or manufacturing process as fast and as cheap as possible. By the way, this flexibility can only be obtained by increasing the complexity of the system; which can be called a drawback on the FMS approach. Due to development of simulation techniques, the new tools helped engineers to overcome this complexity and so increasing the certainty and predictability of the manufacturing systems. In discussion with industrial partners in this project, it is observed that in most cases, the terms simulation and modelling if used together, can cover a wide range of the product development stage, from the part designing in CAD systems, to the process, line and manufacturing simulation. Today, these new approaches lead the industry to the new generation of simulation tools called "Product Lifecycle

[1] Example of processes that utilizing fixture: Assembling, Boring, Broaching, Drilling, Forming, Gauging, Grinding, Heat Treating, Honing, Inspecting, Lapping, Milling, Planning, Sawing, Shaping, Stamping, Tapping, Testing, Turning, Welding, Robotic Spot Welding, Robotic Painting, Robotic Gluing, etc. (1)

Management". PLM[2] softwares are trying to consider all the mentioned manufacturing issues together in an integrated virtual environment. One result of this new looking is diminishing the traditional borders of design and manufacture. By using the integrated databases in PLM softwares the data flow of all units is managed associatively and so the relation and compatibility of the product planning and process planning units are improved. Obvious benefits of using such systems are providing collaborative workflow process, better system integrity and increased data compatibility. (4)

In the traditional discipline the unlimited iterations between the products design units and manufacture planning units is common. These iterations are very expensive in nature as they cost a lot of time, resource and energy. The new approaches decreased these back and forth movements in the field of product design enormously by providing a common environment that all the managerial and engineering teams can access, watch, verify and interact inside in the real time.

Besides, a new platform called "Flexible Plants" recently started by SAAB and Volvo and pushed the approach of flexible manufacturing one step ahead. 'Flexible Plants' means that the manufacturing system must be flexible enough to accept different product plans in the product line at the same time and can refresh itself in minimum time in relation to the product changes (5). This strategy puts the manufacturing engineers in centre of attention as they have to organize and match all the plans in minimum time and also verify that the current order of production is flexible to the changes. In addition, the manufacturing resources also turning into highly flexible components; which have to adapt themselves to different products in the shortest time. In this scenario the design and validation of the fixtures becomes a critical point and puts the new solutions of handling the fixture planning problem in high priority.

1.2 Project Description

The new tools for digital manufacturing has added great not yet implemented potentials for changing processes, modifying the old disciplines, reengineering of work methods and optimizing work roles in production and manufacturing. SAAB Company as one of the leading car manufacturers in using wide range of simulation tools has decided to utilize these tools in order to improve the current procedures used in fixture designing. This project will focus on implementing new work roles/methods based on a new process and plant simulation and validation tool introduced be Siemens, called Process Simulate software to decrease the distance between design, simulation, and validation in the areas dealing with

[2] **P**roduct **L**ifecycle **M**anagement: is a comprehensive information system that coordinates all aspects of a product from initial concept to its eventual retirement (36).

fixtures, which specially used in BIW robotic lines. This will be done by deep looking into software functionalities and applying some customization to gain the most from the software and suggesting new work methods that will use these functionalities.

The project is limited on the current software functionalities without developing any new tool or API, which changes the software structure. In addition, it is limited to be based on an existing ABB fixture part library. The new suggested methods should also comply with the overall workflow standards that currently is using in SAAB, such as being compatible with the Teamcenter technology.

Some articles (6) have divided the interaction of the user with application into three levels. Figure 1 shows these levels and where this project situated.

Figure 1. different automation levels

Interactive: The user will be guided through the design process but there is no automation. Although lots of tools have been implemented to help the user, the application is not deciding about anything and just follows the user clicks.

Semi-Automate: The application is more advanced from the previous level; some procedures will be guided using wizards and dynamic dialog boxes. User will be prevented to take some error decisions by the application internal checking tools. The user and interface moving side by side and helping each other.

Full-Automate: almost everything is automated in this level; proper decisions will be taken by the application and user can help in case of different options. Normally the application moves and the user follow.

Where this project stands is somewhere in the first mid of 'Semi-Automate Level'.

1.3 Basics of Fixture Design

This section will introduce necessary fundamentals regarding fixture design. The aim of this part, is to give a brief overview on fixture design a better understanding of what a fixture is. Also this part is vital in making platform for rest of the project, as it will shows the main elements that project is involved in and specifically how much they can be categorized and classified. Finally at the end of the chapter a new discussion will be opened about the main differences between fixtures regarding their application. The aim of this part is to brighten the points that should be focused on without losing the generality of this research.

Generally, fixtures are categorized as a tool used in production line to assist manufacturing. Among all other tools such as machines, transport devices, cutting tools, and etc.; fixtures are also another type of resources that can be accessed by process planners in order to plan the production sequence. By 'production sequence' or 'production plan' we mean the list and methods of operations, sequences and specifications (type, size, number …) of the resources that should be used to produce a part. Fixtures are tools mainly used to manufacture or process duplicate parts accurately by holding, supporting and locating parts in front of other resources such as milling machines, cutting machines or welding robots (1).

To point the problem, each part demands its own special type of fixture and in addition; the operation that should be performed on the part is also affecting the design of the fixture. This fact made the fixture a unique type of resource. Besides, the fixture is not part of the final product so the shape, material and geometry are not important as long as it satisfies the goals of fixturing. So, the potential for changing exists in the design of the fixture itself which means the potential for iteration between design and manufacture planning engineers.

1.3.1 Fixture Elements

Despite vast range of fixture types and applications, the common building industrial elements of fixtures are certain and limited. This is good news, because a good categorization of these elements can lead to deep understanding of the types and possibility to automating the design process. These elements usually named according to their function in the fixture and then also divided into different sub-categories. Basic category of fixture elements can be seen in Figure 2:

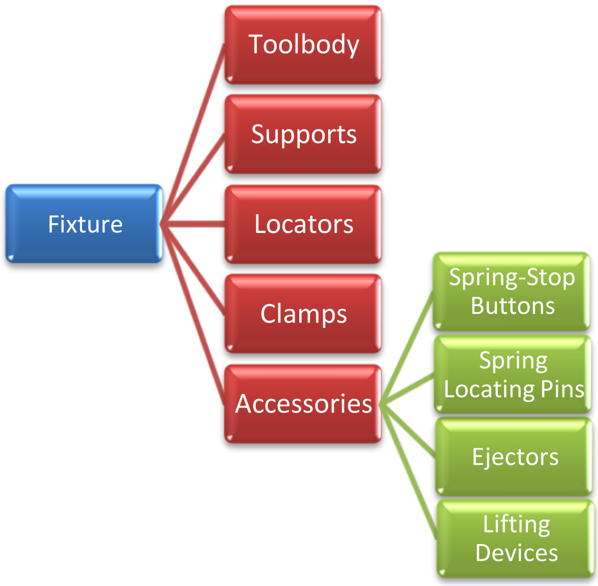


Figure 2. Fixture Main Elements

Toolbody is the base frame that the whole fixture elements will be installed on. It can be made of several materials with different techniques and shapes the main mass of the fixture. **Supports** are elements for tolerating the force. These forces include the mass of the part, machining forces (if any) and clamping forces. Supports are supporting the part against these forces in a proper way and with proper contact points. **Locators** are used for keep the part in the right position. They cooperate with supports to define a unique way of keeping the part in the right position and orientation. **Clamps,** as it is obvious from the name are grasping elements of the fixture. They are insuring that the part will be kept in the fixture securely and tightly during the process. **Spring-Stop Buttons, Spring Locating Pins, Ejectors and Lifting Devices** are optional elements. Their usage is based on the application, cost and size of the fixture.

Each of these groups consists of several sub-groups which in fact are the various types of elements and can be used upon their application and usage. Figure 3 & Figure 4 & Figure 5 show some of these sub-groups:

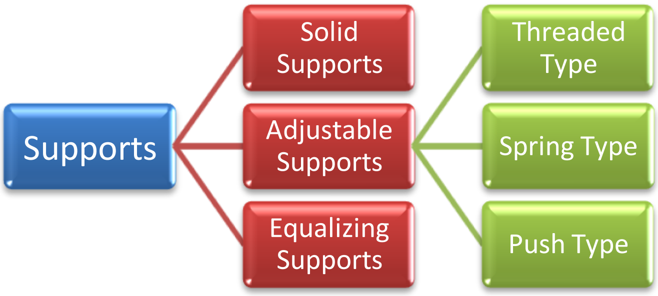


Figure 3. Supports Variations

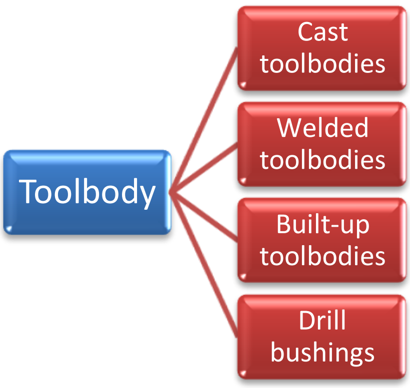


Figure 4. Toolbody Variations

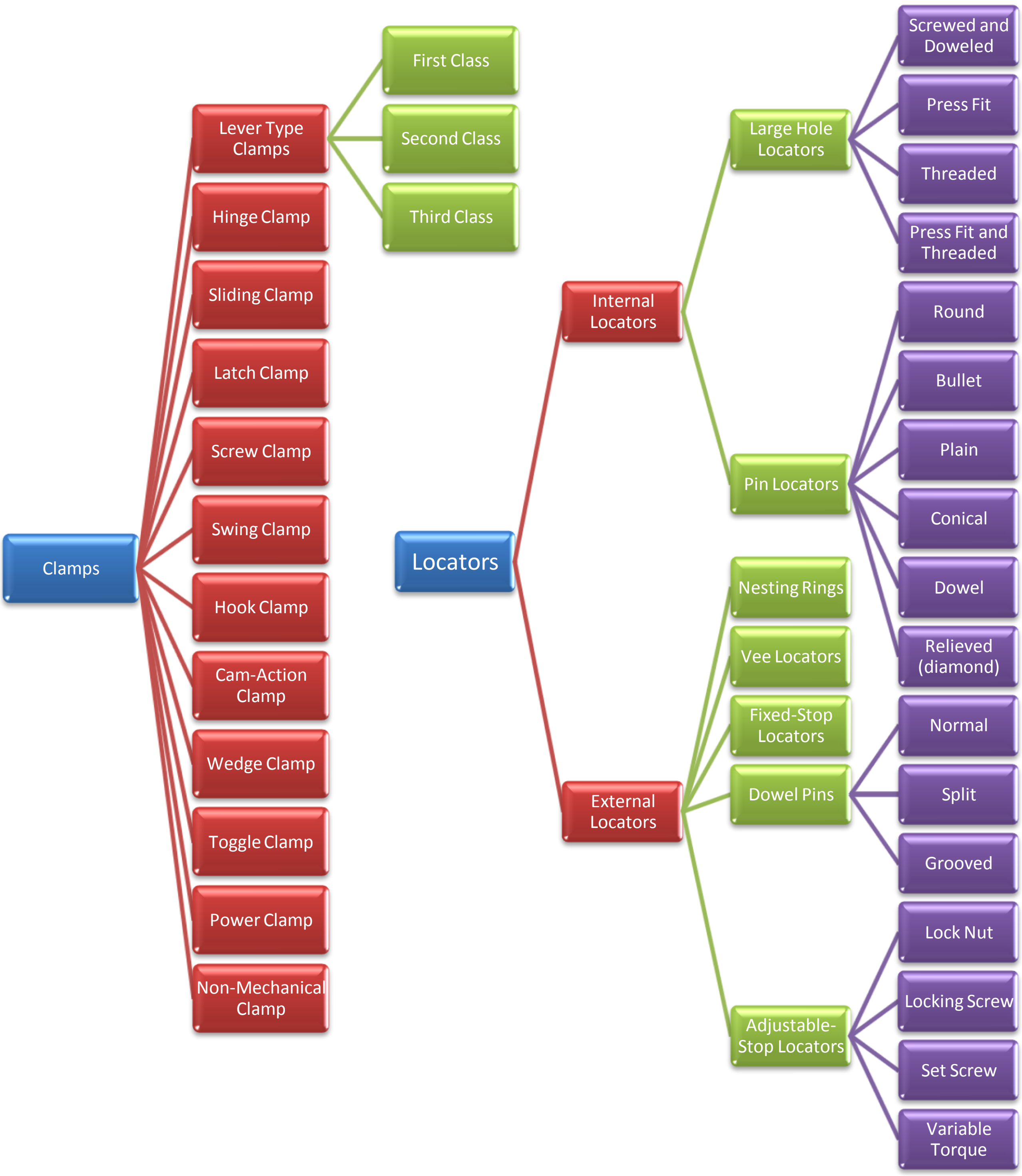


Figure 5. Clamps & Locators Variations

1.3.2 Three-two-one method

Beside the fact that the type and range of the elements used in fixture design is limited, it can be seen that because of the specific aim of each element in the whole design process the order of the design is also somehow predefined. One of the most common methods in designing fixtures is the so called 3-2-1 method. (Figure 6) suggest the following order to design a fixture (7):

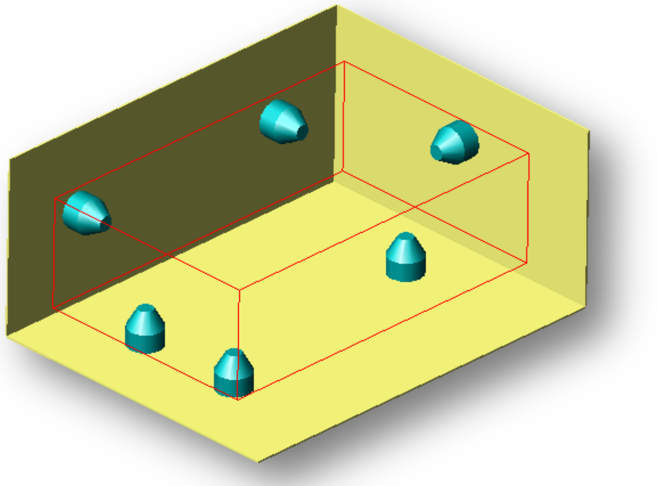


Figure 6. The traditional three-two-one method

1. First designing the Toolbody according to the size and shape of the object.
2. Then supporting the object using **three** supports in base plane.
3. Next, locating the object by using **two** locators in one plane and another **one** locator in the perpendicular plane.
4. Finally holding the object using a clamping device.

In practice the design is not as simple as the mentioned method and lots of considerations should be applied to the fixture. Also as seen before, each category consists of some sub-elements which make the problem more complex. It will be seen soon that all these facts will lead the designers to use standard parts as much as possible and so the need for a part library will come up.

1.4 Modular fixtures & Fixtures for Robotic Applications

The idea of using modular designing[3] is neither new nor specific for the fixture design. Wide range of sciences facilitates using the modular approach because of numerous benefits of that. First of all, it can decrease the cost of change significantly because of the possibility to applying the changes partially just to the desired module and not to the whole system. It also reduces the shut down time of the system during the changing phase and finally it standardizes the design process by facilitating standard parts. (8) Counts some of the benefits of modular designing as follows:

- Modularity creates options
- Modular designs evolve as the options are pursued and exercised
- Modularity makes complexity manageable
- Modularity enables parallel work
- Modularity is tolerant of uncertainty

The modular approach in fixture design appeared in almost the same age of Flexible Manufacturing Systems. In addition, the recent strong intention of using standard parts in the design process accelerated this approach.

To be more precise, by term of "Modular Fixture" two slightly different ideas has been presented. First and older idea is using some ready, precisely pre-holed, or pre-slotted plates and parts that can be fit together according to this holes and slots. Most of the times, a modular base plate is a plate with grid of holes or T-slots that can be served as the mounting point for supports, locators and clamps. In this method, there is not much attention to use standard parts for clamps, supports, and locators. The main idea is more concentrated on easy reconfiguring the fixture layout by changing the location of these primary elements on the base plate. However, because of great flexibility and cost effectiveness, this method is widely used by medium size industries. Figure 7 shows a pre-holed base plate used by IMAO® (9). Other fixture elements such as locators and clamps will be installed on this plate using the precisely measured provided holes.

[3] In the context of systems engineering, modular design — or "modularity in design" — is an approach aiming to subdivide a system into smaller parts (modules) that can be independently created and then used in different systems to drive multiple functionalities. (35)

Figure 7. Pre-holed Base Plate used by IMAO®

The second and newer approach is to use a set of highly standardized and compatible modular parts (Lego like) to build up a fixture. In this method, all the fixture elements from the base plate to clamps are configured and defined before the design process. Therefore, the design process is somehow limited to choosing the proper combination of these elements and joining them together in order to achieve the desired fixture plan. Besides, some new elements such as risers, adaptors, brackets etc. will be added to the previous categories in order to remove any customization in the fixture elements. It means that if a clamp with longer arm or height needed, there will be some standard adaptors or risers to be added to the clamp parts to extend the jaw or to increase the length of the clamp. As a result, a part library should be generated to keep track of all the parts, variations, and families.

This new design method inherits many benefits:

1. Cost reduction due to use of standard elements
2. Limiting the design variety and equalizing the methodology
3. More precise and practical reproduction of the fixture
4. Well managed documentation
5. Mobility and compatibility of design due to use of part libraries

Designing the fixtures for robotics applications is widely increased in previous decades because of the increasing rate of using the robots in the industry.

However, the technological progress that can be observed in other robotic areas[4] is much more considerable in compare to the fixture designing in robotic applications. This can be due to several reasons. First, in most cases comparing to other complexities in usage of robots, the existing fixturing methods are satisfactory and so not in the high priority for change. Second, greater part of the attention to the fixtures for robotic applications is put on the technology of the fixture elements (such as pneumatically actuated clamps) and not on the fixturing methodologies. Third, lack of enough tools to integrate simulation of the robot and fixture together in early stages of design, takes the flexibility away from the engineers. And finally, the basic of the fixture design for robotic application is the same as other areas except that in most cases the robots are not performing any strong force-involved-machining on the parts. Painting, gluing, spot-welding, laser-cutting/welding, assembling and material handling are some of the most common applications that utilizes robots and in none of them strong forces applied to the workpiece from robots. So, from this point of view the fixture design will be even lighter and simpler.

At the other hand, the complexity of robotized processes demands more creativity and challenge for designing their specific fixtures. These fixtures are usually active fixtures (not manually actuated) and also much more crowded by different accessory elements. Also, because of the wide range of movements in robotized processes such as spot-welding, the collision detecting problem is coming to high priority. Combination of these two facts together along with the modular approach converts the fixture designing problem to fixture planning problem. It means unlike traditional fixtures that the force closure plays the vital role in design; in fixtures for robotic applications the collision avoidance plays the first role.

[4] Examples of these areas are time analysis of production line, robots control systems, robots virtual controllers, robot simulation software, manufacturing process planning in the robotic cells, robotic hardware optimization, etc.

2 Project Background

It was mentioned that by the aid of FMS systems the distance between design and manufacturing has been decreased greatly. However, an important issue is that the focus of all these systems is on the part and not the resource. It can be a correct approach because firstly, the aim and output of the system is part production and secondly most of the resources are fixed, ready elements (e.g. robots, containers, tools, transporters ...) that do not need a very complicated design procedure at least by the product manufacturer. Among all, fixture appears as an exception to above while it is a very important resource of the process. In most of the times, essentially the manufacturers have to design customized fixtures for the parts. Worse than that, sometimes a part needs more than one fixture during different manufacturing steps.

Furthermore, in some industries like Body in White[5] manufacturing the presence of robots is very common and at the other hand the part shape imposes very crowded and complicated fixtures to the system. So, the process verification especially collision detection and point reachability validation is one of the main problems that manufacture planners are involved with in this kind of processes.

As a result, now that the industry have the tool of integrating design and manufacture; it would be a great benefit if some methods can be proposed in order to identify and verify the above mentioned problems in early stages and so preventing the back and forth movements in the design procedure both for part and fixture. Meanwhile, changes in design can be rarely applied to the part, as there are many resistances to do that (design restrictions, shape restrictions ...); and so the fixture is mostly the target of these modifications. This is all about what will be investigated in this project: a study about the best methods to design, simulate, verify, and modify the fixture by utilizing modular elements in an integrated environment called Process Simulate.

2.1 Literature Review

A wide range of articles regarding fixture studies such as fixture design, fixture modelling and automate fixture planning was studied during this project. Interestingly, it can be seen that these topics follow a historical line divided into different areas. These areas also include number of major topics that learning them is essential for making a good understanding of the issue. More or less in this project, it has been tried to cover all this line relatively and efficiently. Figure 8 shows the evolution trend of fixture designing and its major topics in each period during years. It also illustrates studied literatures for this project and location of each in this history line.

[5] **Body in White** or **BIW** refers to the stage in automotive design or automobile manufacturing in which the car body sheet metal (including doors, hoods, and deck lids) has been assembled or designed but before the components (chassis, motor) and trim (windshields, seats, upholstery, electronics, etc.) have been added.

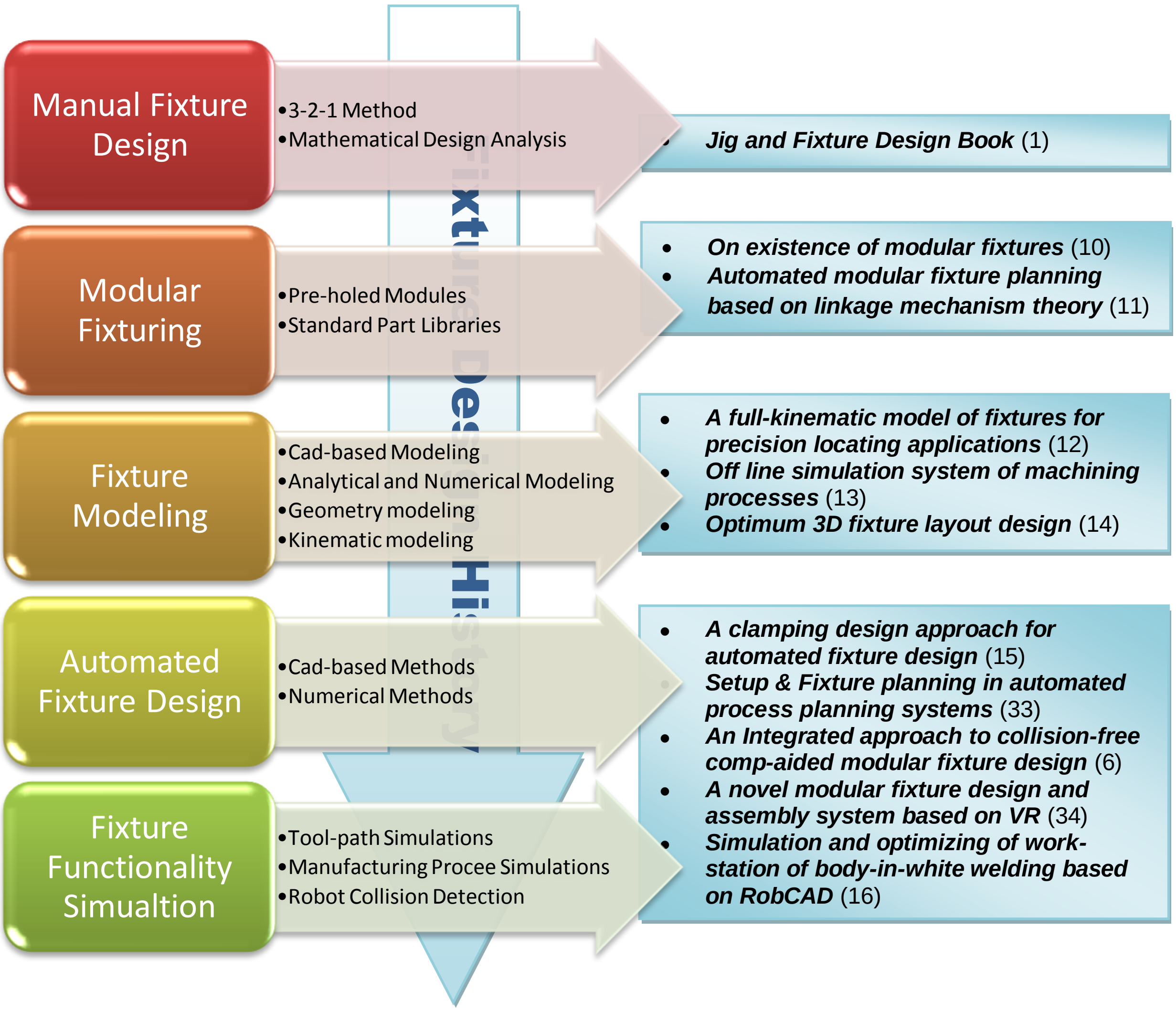


Figure 8. The History Line of Fixture Design

In the first stage of this chart the manual design comes. Obviously, the basics and fundamentals of fixture designing are the main issues in this stage. Most of the topics of this part have been covered in fixture design books and handbooks. **Hoffman** (1) has covered a wide range of these basics and a summary of them has been reflected in the chapter 1.2 and the pre-study report. It can be seen that these knowledge will help to categorize the different fixture elements and it will guide to establish a parametric part library according to this knowledge.

In the chapter of modular fixture design, it has been mentioned that two different approaches exists and although the idea is quite old, still there are topics to be discussed and investigated. Some, like **Zhuang et al.** (10) have tried to investigate the answer to the existence question. Their article tries to check that with a given fixture configuration and a part, does a fixture exist that can hold this part in form closure or not? This can be important because of two reasons: first, in the topic of automated design, it is crucial to know whether a new complex part can be fixtured using existing fixtures or shall design a new one. Second, perhaps the designer has simply overlooked a fixture design that does not require customized components. This can be avoided with algorithms that consider all possible fixtures for a given part.

In the case of modular fixtures with pre-holed base plates, **Wu et al.** (11) have tried to introduce some methods for automatically decide the locator's positions on the base-plate. They assume that the part is supported by the base plate top face and so the main problem is how to locate it accurately on this plane with the aid of dowel pins. Then it uses the linkage mechanism theory to find all the possible holes that can be used to fix a locator on, and called them as candidates. Finally, it utilizes the 'IRC triangle'[6] and 'locator visible cone'[7] approaches to cross out the infeasible candidates and mark the possible clamping areas. There are two main problems for using this method in our specific project: one is that in body in white manufacturing most of the locating points determined by some holes in the part and not the edges (the part is usually sheet metal and so it has sensitive edges). Second, it does not consider the side effecting parameters like the tool or robot path in determining locator and clamp positions.

Fixture modeling can be discussed from several points of view. Depending on what the purpose of the model will be, different approaches can be used. Cad-based modeling, geometric modeling, analytical and numerical modeling, and kinematic modeling are examples of different approaches. The primary objective for modeling of machining processes is to develop a predictive capability of machining performance although most of the times the purpose of modeling is to suggest some optimization methods which is based on the model type. As an example, **Wang** (12) tries to optimize and minimize the tolerance error in fixturing of sensitive parts such as airfoils or tiny electronic parts by proposing a new model that considers the surface geometry of the contacting points. It means that in his model the previous assumption of ideal point contact between the part and fixture improved to a new kinematic chain model in these points. He believes that the previous conventional model, which ignores the underlying surface properties of the locators plus workpiece system, is inherently incapable of capturing the effects of the geometric properties important to accurate positioning of the workpiece. **Deiaba** and **Veldhuis** in (13) proposed another approach which has used finite element method to implement an analytical and numerical model in order to optimize force and dynamic response of the part. They have

[6] Instantaneous Rotational Center Triangle: the triangle whose vertices are the three intersection points between the direction lines of the three locators' reactive forces against the workpiece. See more info on the reference. (11)

[7] The visible cone of a locator is defined as the sector region on the plane of the base plate that consists of all the directions from which the locator can be seen. (11)

also done some case studies on their model and have reached to a very good conclusion:

> *"The results of this case study demonstrate the importance of the fixture configuration to the stability of the process and the geometric integrity of the machined part. It also illustrates how by only modifying individual fixture element locations the response of the system can be altered and driven to a stable zone and the part deflection can be minimized. It is unfortunate that such analysis is typically only carried out after the part has been already in production and many parts have already been scrapped. The current methodology could be integrated into the design process and used to support a more concurrent environment with the end result being a more optimal fixture configuration prior to part production. This will save considerable costs and time."* (13)

Their conclusion can be considered as another support for necessitate of the current project.

In another study (14), **Wang** dividing the fixture design problem into two main parts: 'Fixture Layout problem' and 'Fixture Setup problem'. Accordingly, he defines the former as a task of determining the number, type, and location of fixture elements and the latter as an appropriate fixture assembly designed and constructed from the layout with concerning the collision and interferences between tool path and fixture. He tries to solve the first problem by focusing on efficient numerical techniques for automatically generating, analyzing, and optimizing fixture layout. For this purpose the functions of fixture has been counted in the paper as follows: (a) stable resting; (b) accurate localization; (c) support reinforcement; (d) stable clamping; (e) foreclosure (or total restraint) and (f) quality performance. Then by a matrix called contact matrix he has tried to model these functions numerically and optimize the model by exchanging, adding, or deleting fixture elements and so pursuing an optimal answer. One of the strength points of the paper is concerning localization accuracy (which means the maximum workpiece positioning error) by considering different effective parameters. Positional accuracy of the locators, geometric variability of the workpiece, contact forces generated by locators/supports which reacting to clamping force, contact friction and deformation are considered parameters that is used in the model. However, according to this categorization, the fixture setup is the main concern of current project that is not studied in Wang's paper but practically; can these two problems be solved independent of each other? Alternatively; at least how successful will be fixture layout without considering the fixture setup at the same time, is the matter of discussion.

In the topic of automated modular fixture design, which is the main concern of this project, quite a lot of works has been done. However, the current achievements still have a certain distance from what industries expecting. **Wu et al.** (11) believe the main limitations of the current works are: (a) because the assembly relationships between locators and the workpiece are not described in an analytic way, the current methods of determining location plans are enumerative in nature and thus very time consuming; (b) the current approaches to determination of the side clamping positions are relatively complex; (c) the existing quality metrics about fixture design do not consider the geometry structure of the workpiece and the assembly relationship between the workpiece and locators which also affect the quality of location plans. In addition to above, the following also can be counted: (d) plurality of geometric model definitions standards and file formats avoids proposing a general comprehensive solution; (e) lack of an integrated environment, which can be utilized for connecting to different analytical tools and determining the best solution by considering different parameters.

Kow et al. (6) proposed a rather comprehensive solution to the modular fixture design problem. They have described a CAD-based methodology for developing a modular fixture design system using a solid modeler, integrated with a modular fixture element database. Their study is considerable from different points of view. First, they have used standard commercial software (Unigraphics NX®) to develop their fixture planner module, which has made their work repeatable. Second, they have defined different complexity stages (Interactive, Semi-Automate, and Full-Automate) on their module according to the needed automation, which provides flexibility and functionality and making the design process more efficient. Third, they have used standard part library with parametric design approach. Fourth, they considered the collision problem in their method by importing the cutter location source file (CLSF) into the system during the fixture design process. In addition, they accurately mention that the future trends of their work will be the development of a comprehensive and complete automated fixture design system that integrates the CAD/CAM/CAE tools. According to their model there are six major types of fixtures elements in their suggested part library namely, fixture bases, supports, locators, clamps, adaptors and mounting accessories. In each of the types, there are some families, which comprise a series of fixturing elements of similar geometry but different sizes. Hence, by using parametric modeling, they used a library of single parametric model for each of the fixture element families that reduce the size of the database dramatically. The elements can be picked manually (interactive mode) and/or automatically (semi automate and fully automate mode) and placed on desired point.

In a more specific work, **Cecil** (15) uses CAD model of the workpiece, the tolerance specifications, process sequence, locator points, and design, as the inputs to the system. Output will be automatically identify clamping surfaces and clamp points on a given workpiece using geometric reasoning techniques and designing proper clamp size consequently. However, It should be mentioned that only one type of clamp (strap clamps) has been used in this study.

Finally, **Ma et al.** (16) is the only studied article that points to the simulation of the robots and weld path optimization considering the collision of the robot with the part or fixture. In addition, they are the only who simulated the fixture mechanism and kinematics in a software called RobCAD®[8]. They have used a simplified kinematic simulation of the fixtures in order to measure and optimize the manufacturing time. However, their solution to the collision problem is based on changing the path and not the fixture layout. In addition, the change in the path has been done completely manually in their work that can be a considerable weak point of their research.

2.2 Commercial Products

The optimal target of all the academic works is to provide a powerful background to enhance the solutions for real practical problems that industry faces. So, most of the times, a commercial product is resultant of many academic works. Hence, it has been tried in this project to cover the previous related industrial solutions to the problem. However, the MODFix project that is a running project at present and this project is part of it will be discussed in a separate section.

Some products such as 'Design Modular Fixtures on the Net' (17) which is an online modular fixture design tool cannot be called commercial in fact and they are just a demonstration of some academic works. The module has a very simple and basic tool to design the outline of the part online on the internet and then it will suggest a fixture pattern for that immediately on the screen. However, it can be used to show the idea of the modular fixturing.

However, there are also more industrial ready products in computer aided fixture design; two of them have been investigated during this project. It should be mentioned that none of them has been used directly; the proposed information is based on the existing written documents and reviews which claimed by the manufacturer or their customer. One belongs to 'IBM CATIA V5® Jig and Fixture module' (18) that utilize the parametric design and rule-based template approach to provide consistency and compatibility between the part and fixture design all over the process. The module has been used by number of costumers like COMAU® (19) and BMW® (20) successfully.

The other module called 'NX Jig and Fixture Design Tool' suggested by Siemens PLM Software Inc.® as an integrated module with Unigraphics NX5.0 package

[8] Tecnomatix ROBCAD is the robot simulator system using proved mainly in the Europe automotive industry by Siemens Product Lifecycle Management Inc. ROBCAD offers comprehensive verification functionalities for each application, and shortens time needed for the processes from verification of robot facility to teaching (37).

(21). It also supports parametric design and part library approach. Some of the interesting claimed features are as follows:

- NX Assembly mating conditions allow a new or updated part of the same type to be positioned in the fixture automatically.
- Full associativity between the product model and the fixture can ensure fast, accurate updates.
- NX CAM can use fixtures directly in the machining environment, including fixture collision avoidance and detailed machine motion simulation.
- Welding fixture completes with clamps and welding guns for an automotive assembly.

In the issue of part library approach and modular fixturing there is a world leading Japanese company called IMAO® (22) which developed a complete range of elements used in modular fixturing systems. Their catalogue includes a wide range of locators, supports, clamps and other fixture accessories in different size and shape (9). Their product includes both pre-holed modular and normal fixture elements that can be used based on the application. Figure 9 shows a sample picture from the catalogue.

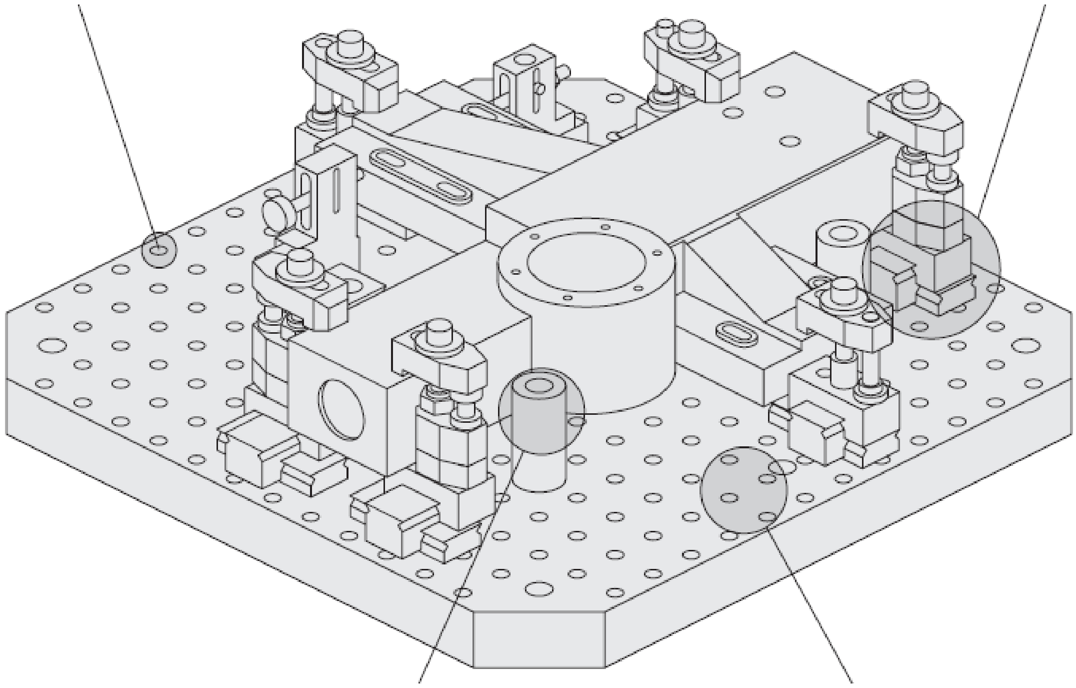


Figure 9. IMAO® Catalogue of Modular Fixture Design

2.3 MODFix project

MODFix project is a contribute project defined by Volvo Cars, SAAB, Högskolan Skövde and Högskolan Väst. The aim of the project is to develop tools and methods to reduce lead-time and man-time with 50% for design and verification of fixtures for Automotive Body in White (BiW) lines during the simultaneous engineering phase (23). The final module will be on Process Simulate® and from this point of view; it is similar to the presented project. The main difference is the present project will just focus on modifying or suggesting new methods, which utilize the existing Process Simulate functionalities, but the MODFix project is much more extensive and based on developing new modules in the software. The main concept of the project is using the standard components for building the fixtures in the Process Simulate environment. These components will have several configurations that are predefined in an administrator module (24) (25). With the use of these configurations, different variations, and fixture families is definable. The main reason for choosing Process Simulate as the target is the capability of software to verify and simulate the whole process including the fixtures in early stages of the design. According to the project definition, the design sequence of the fixture in the new module is as follows:

- Search for PLP[9]: PLP points are determining by the design engineers and transferring to the system along with the part geometry. The first step will be searching and choosing each of these PLPs.
- Select clamp: Next step is automatically choosing a suitable clamp from the library with considering part geometry and other important factors.
- Create fixture reference plane: In this part of the design, the type and place of the base plate should be determined by the operator.
- Different solutions: According to the previous information, the system will design the rest of the process, choosing the needed components, deciding the sizes and positioning them in the proper location.
- In case of collision detection between the robot and fixture one of the below strategies can be selected as a solution:
    - Possibly moving PLP points
    - Moving or reorienting the robot gun
    - Changing or resizing the fixture component
    - Moving the fixture clamp to another place
    - Moving/changing the weld point

[9] **P**rinciple **L**ocation **P**oints are the points which the fixture touches the product and usually determined in product design stage.

3 An Introduction to Tecnomatix® PLM solution

As mentioned in the project background the platform of the project would be the Tecnomatix® solution that is currently used in SAAB factory. This chapter tries to make an introductory to these software packages and shows the merging point between this project and the software. It seems necessary to emphasize the fact that the project focus is on the functionality of the software 'as it is'. For several reasons the project scope defined on not developing any new modules and APIs and just move in the bounds of existing package. For this reason, a thorough research has been made through the software reference manuals to point out any useful functions and combination of functions that can be utilized in the project.

Tecnomatix® PLM solution consists of number of modules that integrates the process design, verification, simulation, and modification. These modules are pointing to a common database called eM-Server and communicating with it. Many works have been done to provide a consistent environment, which guarantees the access, sharing and dividing of the projects among users in a standard way. Figure 10 shows the main modules in the Tecnomatix®. eM-Server is the common base for sharing the knowledge and data between different modules which are working on top of it. Beside eM-Planner, Process Designer and Process Simulate as a core (referred to as eMS); there are some other modules such as Admin Tools, Tab Designer, and RobCAD etc., which are not the focus of this project. However, Admin tools and Custom Tab Designer will be used in the customization part.

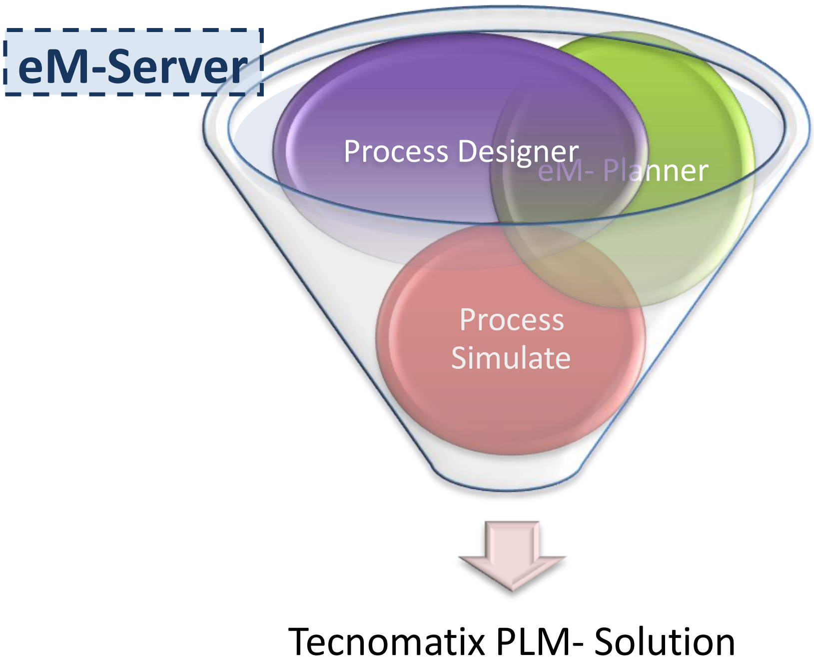


Figure 10. Tecnomaitx PLM Solution

3.1 Process Designer vs. Process Simulate

Process Designer is a digital manufacturing solution for manufacturing process planning in a 3D environment. It is allowing manufacturing organizations to bridge product and process design with integrated authoring capabilities that leverage digital product development. When a new project starts, after transferring the needed data such as 3d part files, assembly structures, product structure and … through a special process called 'collaborative context' to the eM-server, the process planners are starting their job in Process Designer. The main task of Process Designer is to define the structure of the plant, lines, stations, resources and product in a 3D space and relate all these together by several operations to show and verify the flow of products and resources. The key objects in each project are products, resources, and operations. In a car factory, the car body parts are products, the conveyors, robots, fixtures, containers and … are resources; and the instructions of how to assemble a part using these resources are operations. In addition, there is another object called manufacturing feature, which relates the parts and resources. An example of these 'MFGs' is the points that the fixture clamp will touch the part. Figure 11 shows the relation between these basic objects in Tecnomatix solutions (eMS applications).

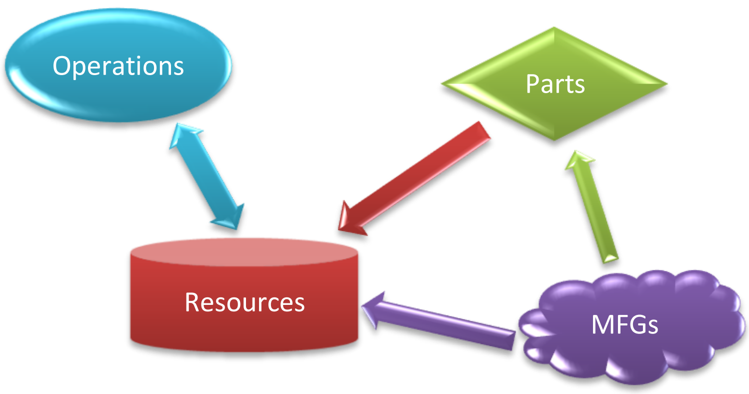


Figure 11. Data structure in eMS

There are number of tools in Process Designer, which help the user to verify and modify the process. Gantt charts, PLP manager, Power Bar and … are examples of these tools.

Usually the Process Designer and eM-Planner are called the process design tools. However, the user needs more tools to verify and validate the simulated process. By Tecnomatix, these tools referred to as 'study tools'. Process Simulate is one of the study tools that are used to simulate the process from different point of views. It provides a unified environment enabling manufacturing engineers to analyze process plans in detail, validate assembly sequences, automatically calculate motion paths, dynamically study collisions between objects and easily access a library of existing resources to improve asset performance. Besides, Process

Simulate provides some unique tools for adding kinematics, digital signals, and sensors to the project. These tools will be used in the project to add new features to the power clamps and provide a simple and efficient way to communicate with and control the fixture via digital signals. Although there are some modeling capabilities in Process Simulate, preference is to transfer the 3D models from another CAD application like NX using the standard accepted formats. By default Tecnomatix applications using '.jt' and '.cojt' formats to transfer 3D data.

3.2 Teamcenter vs. eM-Server & Database File Synchronization

As mentioned before, the project data should be kept in a database called eM-Server. The server will control the access to the data for each user. Each user according to his access level can 'check out' part of the project, work on it and 'check in back' that part via one of the eMS applications. It means that each part of a project can be temporarily borrowed by a user and the user will bring it back with the new modifications. In this way, the application guarantees that each user will load only needed project part (and not the whole project) and two users will not work on the same part at one moment. The structure (schema) of this database, which is determined by the system administrator and in special cases, can be customized and modified. Beside the eM-Server that is an Oracle Database, Tecnomatix uses another place to store the external files referred to by nodes in the eMS database. These external data includes 3D components data, Excel files, kinematic definitions, signal logic relations etc. Each component is placed in a folder that contains all the related data. The directory, which keeps these folders, called **'system root'**.

At the other side, a life cycle data management application (PLM system) called Teamcenter is keeping track of all the design and process data. Teamcenter is a PLM environment supported by Siemens® and widely used in the automotive industry. By default all the design and geometry data of the product is kept in Teamcenter databases. Although Tecnomatix is going to integrate these two databases in near future, it is very important to handle the data suitably until that moment. First of all, the related data has to be uniquely associated to each other in both databases in order to avoid any incompatibility. Also it is important to define certain communication channels between the two and finally the access to the data should be administrated carefully for making the system consistent. In Tecnomatix solutions, the mentioned objectives are satisfying by an intermediate agent called 'Collaborative Context'. Collaborative Context makes the communication channels between the projects in Teamcenter and eMS applications and controls the flow of data. The act of communicating between the Teamcenter and eM-Server is called 'sync. & publish'. Synchronization is data flow moving from Teamcenter to eM-Server and publish is data flow moving from eM-Server to Teamcenter. By default these channels are not always two-ways; certain objects can only be synced and certain objects can only be published. Figure 12 shows the permitted flow of data for different objects between Teamcenter and eM-Server via Collaborative Context.

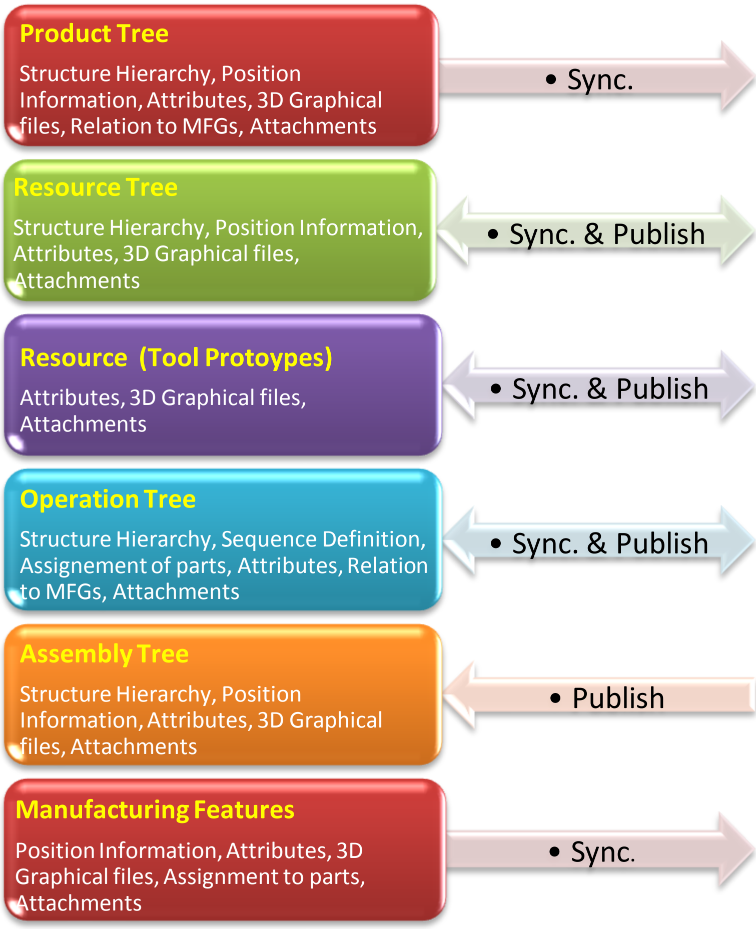


Figure 12. Data Syncronization

4 Design Approach

The following chapter will suggest a general workflow for fixture design and discuss about different methods and functionalities that can be utilized in fixture planning. The desired result is a modular fixture consists of several standard parts. Moreover, these modular components shall contain necessary kinematic behavior wherever needed and shall provide ways to be communicated with, by means of digital actuating signals and sensors. In addition, some tools should be introduced to verify and validate the fixture. The following workflow is actually list of functions, which a fixture designer normally expects to have them. This will depict a picture of what should be searched and implemented. For proposing this workflow, design specifications defined accurately by help of project supervisors in the primary meetings. Next, a comprehensive literature review was done which the results were reflected in chapter 2.1. After summarizing the primary results and getting more familiar with the software using some training materials, the software reference manuals were studied line by line and all useful functionalities highlighted. At the same time, some experiments were done in the software on a real modular fixture sample provided by the SAAB. Then, faced problems were pointed out and discussed with experts in several meetings. Finally, the suggested method tested and implemented by rebuilding the same fixture from scratch using the new workflow.

4.1 General Workflow

The following workflow consists of a general approach to the problem of modular fixture designing and validating. This can be applied to any platform having the potential capabilities and is not exclusively implemented for the tested software. However, this project also reflects the specific methodology adjusted solely for the eMS application family, which will be discussed in Appendix A).

- Project administrator creates new project in Process Designer.
- Different part of the project including Product Tree, Part Library, MFGs, PLPs, Operation Tree, Assembly Tree, Resource Library (Fixture Library exists inside here) is being imported.
- User first checks product parts stability in each station using PLP manager tools. If needed, PLPs can be added or deleted by sending a request to the design centre.
- User starts to design a new fixture by:
    - Picking individual fixture parts from library.
    - Picking collection of related parts (Units) from library.

  - Using fixture units previously used from the archive exists in the library.
  - If needed, users can exchange/delete/move parts to modify the design.
  - Connecting fixture parts together and to the product using connecting frames and PLPs.
- Search tools can be utilized to find a specific fixture part/unit in the library using different search attributes such as length, height, name, type, etc.
- The project is getting loaded in Process Simulate for further studies.
- User connects all the needed signals to the fixture power actuators and fixture switches/sensors.
- Other resources such as robots are getting added and the whole station is being simulated.
- In case of detecting collision between robot, fixture, and product it can be solved by:
  - Moving fixture parts
  - Changing fixture parts
  - Try different configurations
  - Moving PLPs
  - Changing the Path
  - Changing/moving the gun
- User publishes the new fixture for other project members to do further studies.

Finally, Figure 13 shows the suggested dataflow for modular fixture design:

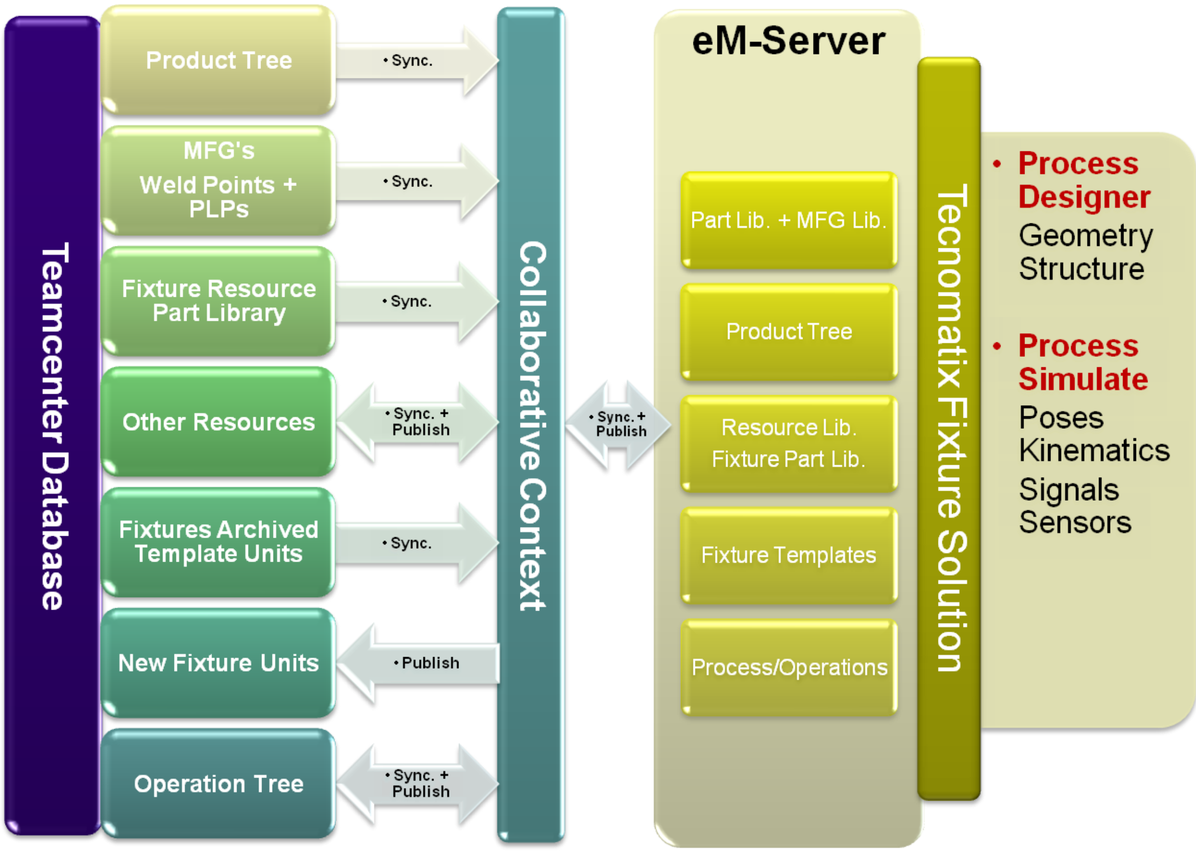


Figure 13. Fixture Design Dataflow

4.2 Part Library Structures for Fixture

In a modular design approach, part libraries and search tools are somehow the most important part of the idea. As the user will face huge number of standard parts, there should be some efficient ways to categorize parts and provide some tools to search among them. Furthermore, there are number of parts that always used together and make commonly used collections. Accordingly, most of the times the user is not starting from a scratch and desires to use this ready to use collections. Therefore, in addition to the part library, also there should be some ways to store and reuse these collections.

Several approaches for the part libraries investigated. The structure of part library is highly depends on the current design standards of the factory. Different factories select different standards for designing the parts. Two distinguished technique in this issue are 'parametric' and 'non-parametric' approach. In parametric approach, for each family of parts there is a single CAD file which defines the geometry of that family by means of several variables to specify the design values (such as size of different sides of the part). Beside this file, there also exists a table, which defines these parametric values for different family members. By using this method, there is no need to store separate files for each of similar family members that just have difference in size. At the other hand, the non-parametric method, stores geometry file for each part independently so the size of the library is increasing significantly but at the same time the design flexibility is also increasing. Each of these methods has their own benefits and weaknesses and as mentioned, their usage is highly dependent on the other data handling standards in a factory. In this project, the second (non-parametric) method will be used as a predefined limitation to the project.

The suggested library for this project has three main branches: **Fixture Parts, Fixture Templates, and Fixture Archives.** Each of these main branches can have several sub-branches. For example the Fixture Parts, will be divided to several other branches based on the fixture part type. Examples of these branches are Brackets, Spacers, Risers, Base Plates, Switches, Fingers, etc. The Fixture Templates are collections. They consist of several fixture parts located beside each other and build a **'Unit'.** Each unit has a name and categorized in some sub-branches based on its usage in the fixture. Examples of these units can be clamp units, sliding units, locating units, resting units, etc. Finally, the fixture archive branch is a collection of all currently used fixtures. Sometimes building a new fixture based on an existing fixture in the archive is much easier than starting from zero point. Figure 14 shows an example of this part library structure.

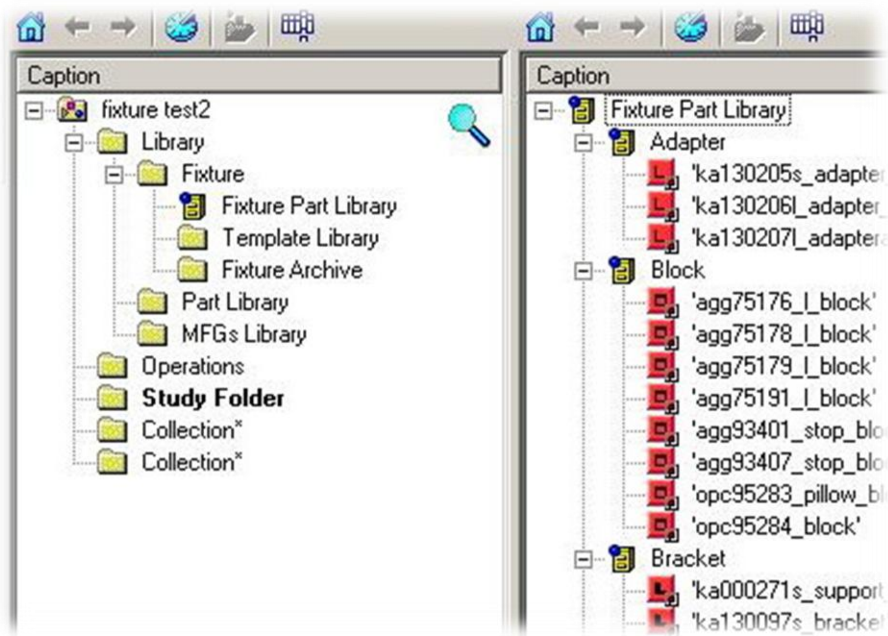


Figure 14. Fixture Part Library Structures

A very important point in this approach is that there is a big difference between the Fixture Parts branch and the other two. The 'fixture parts' is a real library that keeps the prototype of each single part. When the user picks a part from this branch, an instance of the prototype will be placed in the project. Technically this instance is a pointer to the part prototype in the library. Therefore, whenever the part in the library changes, all the instances will automatically reflect this change.

On the other hand, the 'fixture template' and 'fixture archive' are not a real library. They are simply collection of nodes, which points to some parts in the first branch. This means whenever the user wanted to use them in the project, a copy of these units should be placed in the project (using the CTRL key while picking). This is because the templates are not independent members; they are just collection of pointers to the real part library. Therefore, any future change to the template or to the picked objects in the project will not reflect to the other side.

Checklist of build up and use Fixture Library Structure:

Administrator level:

- Building the nodes structure in CAD application
- Importing the synced library using Collaborative Context interface
- Assigning the types to the library members
- Populating the templates and archives

User Level:

Building the fixture by:

- picking the parts from the library or
- copying units from the templates/archive library
- if needed, exchanging the parts inside the units in the project using library

4.3 Attributes and Power-bar Functionality

For enabling the user to search in the library for a specific part, search functionalities should be investigated. When the library parts are importing to the project, usually some attributes is attached to the part. These attributes can be different things, part name, part number, author, and design parameters such as length, height, material and ... therefore by defining new attributes and setting proper values for them the user can later use them to decide which part shall be used in a specific case. Although Process Designer accepts setting new attributes inside it, it is more desirable to import the attribute values from the CAD centre to keep the compatibility as much as possible. Furthermore, beside all the parts that can have their own attributes specifically, the nodes (units) can also have attributes. This will help the user to search for the certain unit inside the template library. So, one important part of the system administrator task is to smartly choose the attributes type and setting their values. For example although each part in clamp unit can has its specific length, the clamp unit also can have an effective length by itself to help the user decide which unit should be selected without the need of looking inside the unit and considering individual member lengths.

Power-Bar is a tool in Process Designer which can search in the project for special features. Without having a powerful search tool, modular design is not possible as the parts cannot be picked by the user manually because of huge number and vast range of objects in the library. By default Power-Bar is capable of searching for certain objects such as PLPs, Clamps and Weld-guns in a predefined search scope using different attributes. According to the type of attribute different search variables is selectable. Also, the power-bar let the user to combine different attribute search results to do more advanced searching in the project

By applying some customizations it can be used to search for fixture parts in the library. These customizations will be studied later in the customization summary chapter.

Checklist of using attributes and search tools:

Administrator level:

- Define and create different attributes according to the object type in the CAD application
- Setting the values of the attributes
- Defining the same attribute fields in the eM database
- Customizing the Power-bar by adding the fixture new objects and their searchable attributes and the search scope.
- Adding new tab in the object properties window to simplify fixture specific attribute handling

User Level:

- Using the power bar and properties window to search and determine suitable fixture parts/units

4.4 Assembly Methods

In a modular approach, making the connections between part modules is very important. After picking the components and/or collection of components (units) from library or archive and putting them in the work space, next step is to connect these objects to the other objects already existed there. Although this connecting procedure is not a complicate process, it can be time consuming. So, it is necessary to introduce some useful tools/methods in order to do this with minimum clicks. In addition, some modular components have several assembling configurations. For example, a clamp can be installed on its base by different orienting angles (Figure 15). So the designer is expecting to have options for choosing these different configurations.

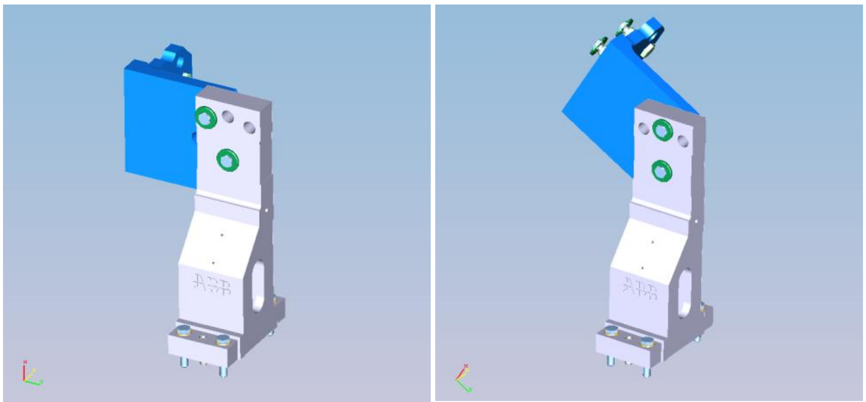


Figure 15. Different Assembling Configurations

Unfortunately, the existing placement tools of Process Simulate and Designer are not very powerful. Especially they are not able to specifically pick custom points using selective snaps (such as centre points). Besides, in many cases the important object points cannot be referred by any snap points. In addition, for connecting parts to each other, picking just a point is not enough, orientation angle is also important.

Therefore, a good approach is to define some connecting frames for each element. These frames not only show the point that objects should be connected to their precedence but also specify the orientation of two objects in relation to each other. In addition by defining several frames, different possible configurations can be hinted to the user and can be easily chosen by him. For making it easier, these frames can be divided to two categories: input frames and output frames. In this way the user knows how to connect the part to it's before and after parts. The two can be identified by choosing different names or colours.

Although it is a very good and may be the only good solution, it suffers from a problem. CAD applications such as NX does not supporting the standard frame in the '.jt' file format. So these frames cannot be added to the parts in CAD. It seems that this is due to some incompatibilities between '.jt' and '.cojt' formats between different platforms and a good long term suggestion can be to ask UGS for adding this feature. The frames can be added to the parts in Process Simulate but does not saved in the part 3D file. So a not very satisfying solution is to open each part once in Process Simulate, adding the needed frames and save it back to the library.

Another solution can be to use the temporary frame capabilities in the Process Simulate. While using the placing tools, there is an opportunity to define a temporary frame by several methods and pick the point according to this frame. The methods include making frame by 3 points, making frame by 6 values, making frame by centre of circle and making frame in between 2 points. So, instead of adding the frame in CAD application it is possible to draw some guide lines in order to help the user make the temporary frame according to these lines. It is one step ahead but not very satisfying as lots of lines in the work space makes it very crowded and seems very confusing. Also one another solution can be to make a 3D component looks like a frame and add it to the part assembly as an independent component in the CAD application which can hint the user to build the temporary frame upon this object. The same problem exists also for this solution unless these frame-looking objects can have name and so can be filtered by name to make the workspace less crowded.

Checklist of using frames for assembling parts:

Administrator level:

- Determine the input and output connection points of each part by considering different configuration for each part
- Add input and output frames whether in CAD application by virtual frame symbol or in Process Simulate by create frame command

User Level:

- Connect the parts together using the 'relocate' command
- Get help to select the points by clicking on the existing frames or by making temporary frames

4.5 Adding Kinematics to the Fixture

Fixtures are situated in the category of the resources, which usually called active resources. It means that like robots, there are moving parts inside each fixture. These moving parts can be either actuated by human or other source of powers such as pneumatic, hydraulic, and electrical. As mentioned before, most of the fixtures that are being used in BIW and Robotics industry are in the second type, which usually called power fixtures. As an important part of this project, investigating on the kinematic properties of the fixtures seems essential to build a realistic model of the fixture. Furthermore, this kinematic behavior of the fixture is needed in simulation of the process. Therefore, this part of the project will search for best solutions in Process Simulate to add kinematics to the fixture elements.

By default for defining the kinematics, in Process Simulate there is a menu called kinematics. The main functionality of this toolbar is to manage the kinematic properties of the movable objects. Kinematic chain is the simplest element of this type of objects and consists of at least two links and one joint. Links are solid bodies that are moving relating to each other and joint is an axis that defines this relation. Simply joints can have two main types: revolute joint, which defines rotational movements, and prismatic joint that defines prismatic movements. Example of these kinematics joints are robots. By definition robot is a collection of several kinematic chains that also have a target coordinate system called TCPF. In the fixtures, clamps and sliding parts are usually the ones, which need kinematic behavior. Figure 16 shows a sample clamp in two different positions.

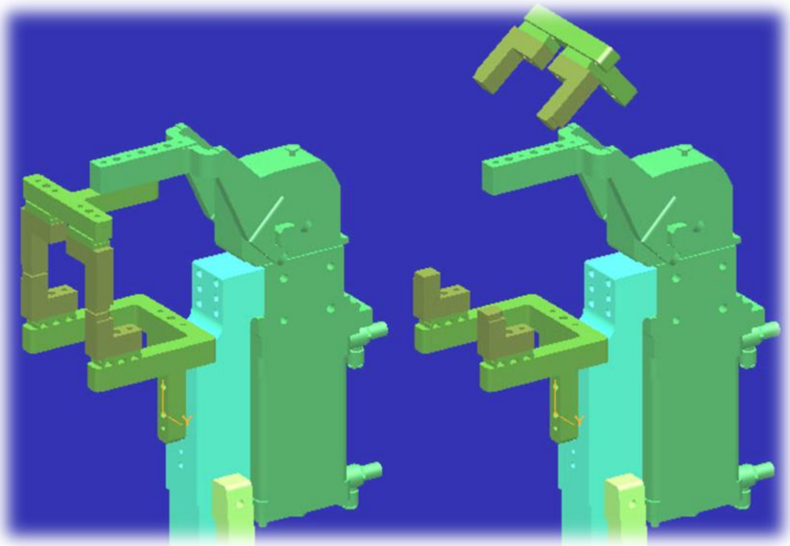


Figure 16. Different Kinematic Poses for A Clamp

A very important fact in Process Simulate kinematic part is that the moving parts should be situated in one file. It means that a single 3D file should contain all the moving objects of one kinematic chain. As investigated, this is not possible in our project because of several reasons:

- First, each clamp or sliding unit is consisting of several single parts, which are picked individually from the fixture resource library. This means that the parts of each clamp are in different files and no kinematic chain can be defined between them.
- In the sample SAAB fixture file, which referenced in the project, the moving arm of the clamps is united with clamp body and although they are in one file, it is not possible to define kinematic chain between them. This is due to the default transferring properties of the JT files from the NX to the Process Simulate. Be default, assembly files are situated in separate part files and NX cannot save assembly in a single JT file although this feature is supported by JT format.
- Even if all the clamp parts can be situated in one file this is not the solution. The user should have the possibility to simply exchange these parts with another size or configuration, which is not possible if they were in one file.

Many different methods examined in the project to see if there is any single way to overcome this problem and finally two of the Process Simulate functionalities mixed to make a satisfying solution out of it.

First, the kinematic chain can be defined by a kind of trick. In the kinematic editor command, the user should define the links and associate the 3D parts to the links. Then a joint should be defined between these two links for specifying how they should move related to each other. The solution is to define the first link as normal and select the clamp body for the 3D part of the first link and then defining the second link without specifying any part to it. This will result to two links, a real, and a virtual link. Process Simulate doesn't care if the links are really exist in the outer world and the user can simply continues and define the joint which is the rotation axis of the clamp arm between these two links. The speed and acceleration of the clamp can be also defined in the joint definition dialog. Now, at least we have a kinematic chain, which can be moved and simulated although there is no graphical movement in the screen.

- **NOTE 1:** for applying above changes to the clamp part, the 'set as modeling space' should be activated and after applying the modifications, the clamp should be 'saved back to the library'.
- **NOTE 2:** in future extension of the work, the clamp file should be modified and the arm and body should be separated (but in the same file). Bu doing this a real kinematic chain can be defined between these two moving part.

The next step is to add poses to the moving part. For example for the clamp, poses 'open', 'semi open' and 'close' can be defined according to the joint value of the clamp. These poses will be used later to add actions to the clamp which is necessary for actuating the clamp by signals.

- **NOTE 3:** never delete the default pose called home in the pose editor. This will cause the system to crash! It seems there is a bug in this part of the application.

A very important benefit of this solution is that the kinematic definition of the base part (power clamp pneumatic unit or pneumatic cylinder or ...) will be saved with the part in the library. So the above procedure should only be done by administrator just once and just for the base moving unit object (and not for all the moving parts attached to it). This property makes the kinematic definition portable through the library and templates.

For adding a real movement to the other objects of the clamp unit, there is a set of commands called attach and detach. These commands used to attach or detach objects to each other in one way or two ways. After attaching the objects the attached objects will follow the movements of their parent. The trick is to attach the entire moving entities of the clamp to the mentioned virtual link of the base clamp. By doing this all the desired entities will follow the second link of the clamp (which is not really exists) and that is what we are searching for. For making the procedure simpler, placing all the moving parts in a 'compound fixture unit node' is suggested (for example, this node can be called upper arm). By doing this, it is only sufficient to attach just this node to the virtual link and there is no need for selecting all the moving entities one by one. Figure 17 shows a better illustration of this solution.

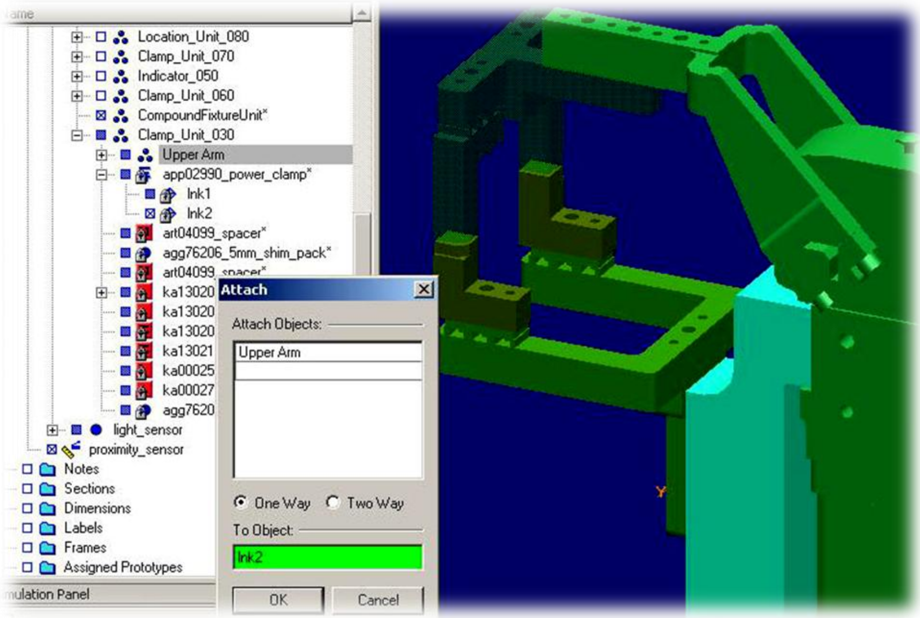


Figure 17. Kinematic solution for fixture units

The only disadvantage of this solution is that the 'attach' command is working in the study level and so the attachment information is not saved in the object or the object tree. This means that when the user inserting new or existing fixture units from the fixture library or template library the attachment information will not be transferred to the new fixture although the kinematic definitions will be transferred. So, after completing the new fixture, one task of the user will be to attach the moving nodes to their parents.

By applying this solution, the kinematic definition will be in the administrator level tasks and attaching the parts will be in the user level tasks.

- **NOTE 4:** a very good suggestion to the UGS can be to ask for adding this feature to the tree level and not only the study level.

Checklist of adding kinematics:

Administrator level:

- Switching to modelling space for the parts which needs kinematic definition

- Defining the links (one real and one virtual)
- Defining the joint
- Setting the joint speed and acceleration
- Setting the joint limits
- Defining poses according to the joint values and selecting proper names for them
- Copying the part back to the library
- Defining the 'moving entities' compound node in the template and put all the moving entities in this node

User level:

- Loading the study
- Attaching the moving entities node to the link2 of the base object.
- Updating the server

4.6 Adding Signals and Actions to the Fixture

Adding the kinematics functionalities to the fixture parts is a big step toward realization of the fixture behaviour but definitely not big enough. In practice the process engineers doesn't care much about the individual movements of the elements in the fixture. They prefer to see the fixture as a whole device which can be communicated through standard methods. It means that after the power devices in the fixture adjusted once, there is no more need to look inside the moving parts and afterwards they should be actuated by the help of controller devices. Furthermore, most of the times it is necessary to make the controllers aware of the current situation of these devices like any other device in the plant. To be more accurate, there should be some signals (fixture inputs) to actuate the power devices in the fixture as well as signals (fixture outputs) to send the current situation of the devices back to the controller. Usually the latter happens by installing different kind of sensors in the device. Figure 18 shows a sample power clamp and its input and output signals.

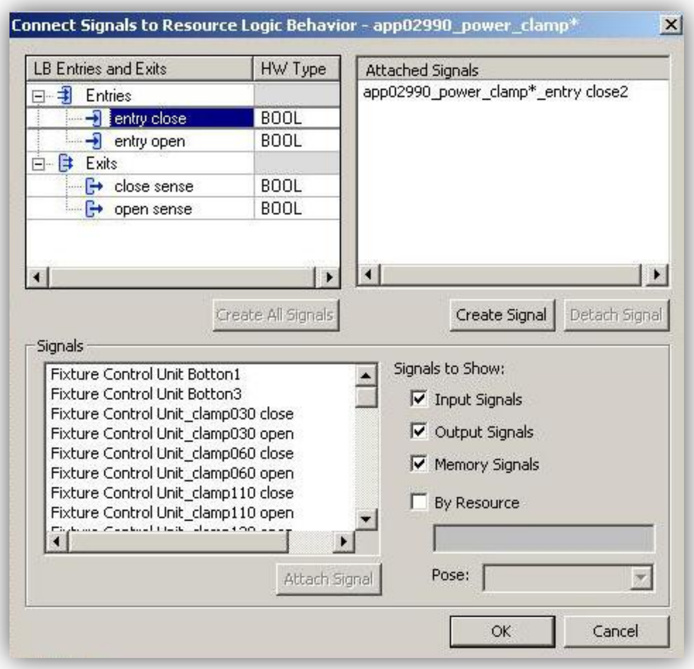


Figure 18. Signals

In practice, each fixture has a controller which controls all the units inside that fixture. This controller itself will be controlled through the central PLC unit in turn. So, when adding a new unit to the fixture, it is enough to install the unit and attach its connections to the controller and program the controller to use these new signals. The project will try to implement exactly the same scenario in the simulation environment.

Before adding the signals, some preparations of the units are necessary. First of all, some actions should be defined for the moving units. These actions will move the units from one state to another. For example in a clamp, if the clamp has two states (open - close), there should an action that moves the clamp form close state to open state and an action to do vice versa. Moreover, there should be some sensors to sense the current state of the clamp and send out proper signal for that.

Different methods considered for implementing the above requirements. Firstly, for defining the movements from one state to another, there are at least two solutions.

One way is to define some device operations. Device operations are kind of an operation which moves specified device from one state to another. Most of the times a non simulated operation which doesn't have any special action inside shall be added to the start of the device operation in order to define a start condition for the device operation and then this group will be activated during the simulation by calling an device start event. Although at the first sight, this solution works fine, it suffers from a very big problem. Naturally, the intention is to store these standard movements for each of the fixture units in the library for further use. The user shall not redefine all these operations every time he is going to use a unit in a new fixture by any means. So, the operation itself should also be stored in the library. Obviously, resource library is not capable of storing operations. Even the template library is not a suitable place for that because then there will be a need to add an operation library and keep track of which operation belongs to which device that is not an easy task at all.

Second solution is referring to a unique functionality in Process Simulate advanced simulation panel. The condition of applying this functionality is as follows:

- Object should be in modelling space
- Some kinematic chains should exist for the object
- Some poses other than 'Home' should be defined for the object according to its kinematic joint values

If the above conditions satisfied, by right clicking the object in the advanced simulation mode a command will appear in the context menu called 'add logic block to resource'. Normally, in the logic resources[10] there is a button called 'Action' which is not active by default and uniquely can be activated by the mentioned method. By using this method the administrator is capable of adding some actions such as open/close and some actuating signals which will run the defined actions to the part itself. This data will be saved as part of the resource to the resource library and can be used accordingly in any new project without doing any extra effort. This solution seems to satisfy all the desired objectives as: the actions can be defined to move the fixture device from one state to another, the actions data will be saved inside the resource and so is portable; and some input signals can be defined which will actuate each action.

[10] A logic resource is a resource that has no 3D representation or kinematics behavior. Instead, it contains a defined logic behavior with respect to the control system, derived from one or more specified inputs and outputs in an equation or formula (28).

Similarly, for adding the sensor data to our devices there are several solutions. By default Process Simulate supports photoelectric, proximity, property and joint value sensors. Each of these sensors can be used in the fixture units to send information to the fixture control unit. Proximity and photoelectric sensors can be used to sense presence of the product in the fixture and joint value sensor can sense position of the moving fixture components. Although the joint value sensor can be added to the fixture devices to send open/close state of the clamps, a better solution can be used for this issue. After adding the actions to the logic block of the moving resources, in the 'Parameter' button of the resource logic block dialog box, a very useful parameter called 'joint value parameter' can be utilized to perform similar behaviour as the joint value sensor. This parameter plays exactly the same role as the joint value sensor without any need to add an extra sensor to the resource. The parameter can be set in the dialog box to be activated whenever the clamp or any moving element reached a specific state or position and the output result can be send to any signal in the resource logic block. Figure 19 shows a sample logic block used for a clamp.

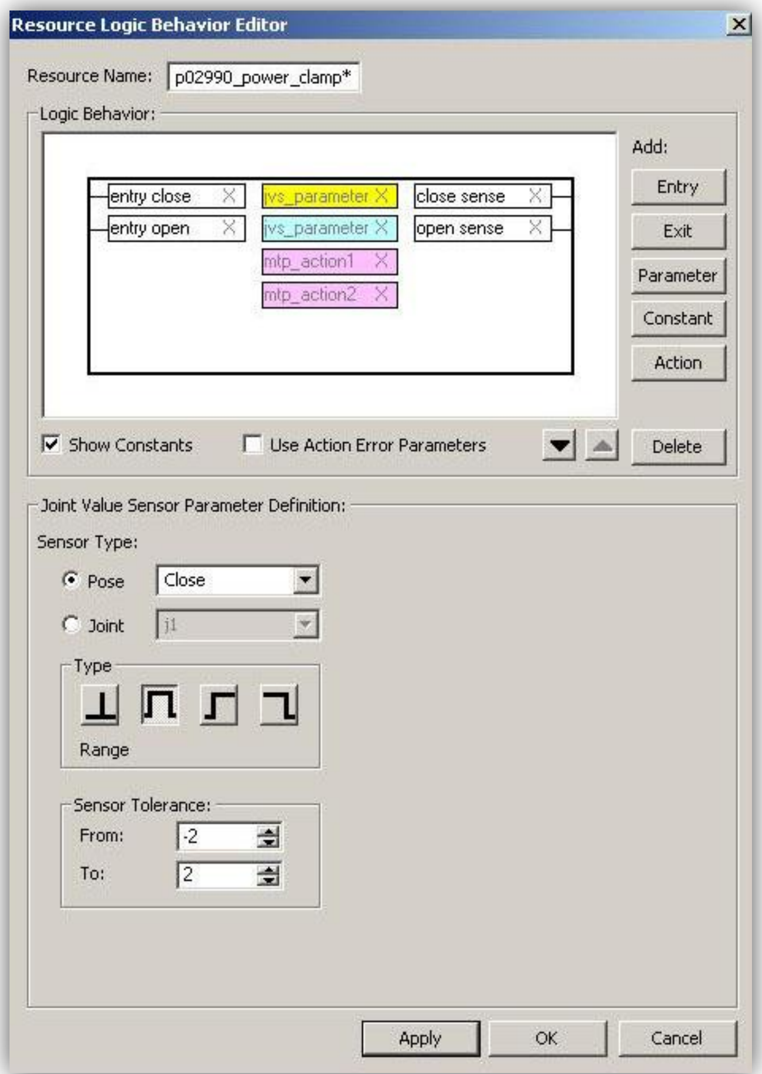


Figure 19. Logic Blocks

- **NOTE 1:** rename the entries and exits to a meaningful name for better handling these signals in future but never rename the inside parameters and actions or the software will crash!
- **NOTE 2:** the proposed solution is the only one which will work independent of any operation. The other ways to add sensors can only be used inside the operations.

Checklist of adding signals and actions:

Administrator Level:

- Getting sure that the desired resource have the kinematics and poses defined beforehand
- Switching to modelling space for resource objects who needs signal functionality
- Adding logic block to the resource object
- Defining the device inputs (entry) and outputs (exit)
- Defining 'actions' and setting their start condition signal
- Adding joint value parameter according to the poses
- Attaching the joint value parameters to the device outputs
- Creating wires for input and output (generating signals for the device)

User Level:

- Connect the fixture unit pre-ready signals to the central control unit
- Program the central control unit how to handle the input and output signals of the device

5 Conclusions

The present project is a single step to blur the traditional borders of the design and process planning. It showed obviously that philosophy of design can be modified and reconsidered as the today simulation tools can integrate design, manufacture, and process planning.

It was seen that the fixtures as an important resource in manufacturing, can be planned and designed simultaneously with process planning. Furthermore, project approved that modular approach can be considered as a very effective solution in flexible manufacturing systems such as Robotized environments. It has also been showed that in modular design, searching and manipulating tools are in high importance. Providing some efficient and user-friendly methods to handle these two tasks can decrease the lead time greatly.

This project also confirmed the idea of task integration. It showed that PLM solutions if structured well; can be a good tool to improve the system productivity. As seen in the project the tasks of fixture designing, fixture validation and process planning, integrated in a single environment. This improved the system compatibility.

Process Simulate as a testing base, proved to be a suitable platform for these kinds of integrations. The system structure founded to be stable, flexible (good ability to accept customizations), and powerful. In few cases (databases synchronization and file format compliance), more investigation on the system compatibility with the current PLM tools seems necessary. For boosting the application power, implementing of extra APIs is highly suggested. Besides, some functions such as 'attach' command and 'frame' creation, if enhanced, can improve the project a lot.

One further works of this project can be investigating on testing different CAD structure approaches. For the start, Parametric and non-Parametric methods can be compared.

Another improvement can be adding the possibility of searching for advanced attribute combinations. As an example, the possibility to define formulas containing different attributes and make the search base on them.

References:

1. **Hoffman, Edward G.** *Jig and Fixture Design.* fifth edition. New York : Delmar Learning, 2003. ISBN: 1-4018-1107-8.

2. *Flexible Manufacturing Systems - A Review.* **Sims, R. B.** 1983, Electronics & Power, Vol. 29, pp. 809-812. ISSN: 0013-5127.

3. *Manufacturing flexibility: a literature review.* **Toni, A. De and Tonchia, S.** 6, s.l. : Taylor & Francis, 1998, International Journal of Production, Vol. 36, pp. 1587 - 1617.

4. **Arena PLM.** Why PLM? PLM Software Arena Solution. *PLM Software Arena Solution.* [Online] Arena Solutions Inc. [Cited: September 23, 2008.] http://www.arenasolutions.com/why/plm/.

5. **UGS, ABB Robotics, Saab Automobile AB, Volva Car Corporation.** *MERA ModFix Project Proposal.* [Pdf Document] 2007-01-24.

6. *An Integrated Approach to Collision-Free Computer-Aided Modular Fixture Design.* **Kow, T. S., Kumar, A. S. and Fuh, J. Y. H.** London : Springer-Verlag London Limited, 2000, Int J Adv Manuf Technol, Vol. 16, pp. 233-242.

7. *Flexible Fixturing for Workpiece Positioning and Constraining.* **Shirinzadeh, B.** 2, s.l. : Emerald Group Publishing Limited, 2002, Assembly Automation, Vol. 22, pp. 112-120.

8. **Baldwin, Carliss Y. and Clark, Kim B.** Modularity in the Design of Complex Engineering Systems. *Complex Engineered Systems.* s.l. : Springer Berlin / Heidelberg, 2006, pp. 175-205.

9. **Imao Corporation.** Standard Fixturing Components. *Imao Corporation.* [Online] [Cited: April 10, 2008.] http://www.imao.co.jp/english/files/whs-004e-mfp.pdf.

10. *On the Existence of Modular Fixtures.* **Zhuang, Yan, Goldberg, Ken and Wongs, Yiu-Chung.** San Diego, CA, USA : IEEE International Conference on Robotics and Automation, 1994. Proceedings., 1994. Vol. 1, pp. 543-550. ISBN: 0-8186-5330-2.

11. *Automated Modular Fixture Planning Based on Linkage Mechanism Theory.* **Wu, Yuguang, Gao, Shuming and Chen, Zichen.** s.l. : Elsevier Ltd., 2008, Robotics and Computer-Integrated Manufacturing, Vol. 24, pp. 38–49.

12. *A Full-Kinematic Model of Fixtures for Precision Locating Applications.* **Wang, Michael Yu.** Hawaii, USA : IEEE, 2001. Intenational Conference on Intelligent Robot and Systems.

13. *Offline Simulation System of Machining Processes.* **Deiaba, Ibrahim M. and Veldhuis, Stephen C.** 5, s.l. : The Franklin Institute, August 2007, Journal of the Franklin Institute, Vol. 344, pp. 565-576.

14. *Optimum 3D Fixture Layout Design.* **Wang, Michael Yu.** Hefei, P.R. China : IEEE, 2000. 3rd World Congress on Intelligent Control and Automation.

15. *A Clamping Design Approach for Automated Fixture Design.* **Cecil, J.** s.l. : Springer-Verlag London Limited, 2001, Int J Adv Manuf Technol, Vol. 18, pp. 784–789.

16. *Simulation and Optimizing of Work-station of Body-in-White Welding Based on RobCAD.* **Yongkang Ma, Hui Li.** Harbin, China : IEEE, 2007. International Conference on Mechatronics and Automation.

17. **Brost, Randy C. and Goldberg, Ken.** Design Modular Fixture On the Net. *Fixturenet.* [Online] 1997. [Cited: Febrauary 10, 2008.] http://memento.ieor.berkeley.edu/fixture/.

18. **IBM Co.** *Generative tooling for automotive jigs, fixtures, molds and dies.* [file] s.l. : IBM, 2006. PLBEE01083-3 (06/03) HW.

19. —. CATIA V5 supports innovation and collaboration at COMAU. *IBM PLM Solutions.* [Online] IBM. [Cited: March 5, 2008.] http://www-01.ibm.com/software/success/cssdb.nsf/ CS/SDUY-6AYARF?OpenDocument&Site=corp&cty=en_us.

20. **Engstler, Dr.-Ing. Matthias and et al.** BMW Group Vehicle Body Division takes CATIA V5 on board. *CAD-CAM Engineering Magazine Report.* July 2003, 7.

21. **Siemens Product Lifecycle Management Software Inc.** NX Jig and Fixture Design Tool. *Siemens PLM Softwares.* [Online] SIEMENS, 2008. [Cited: February 12, 2008.] http://www.plm.automation.siemens.com/en_us/products/nx/tooling/jig_fixture.shtml.

22. **Imao Corporation.** Home. *Imao Corporation.* [Online] 2007. [Cited: April 10, 2008.] http://www.imao.co.jp/english/.

23. **Hörfelt, Hans.** *Modular fixtures for Body In White lines (Modfix).* [Power point presentation] s.l. : Volvo Cars, 2007.

24. **Heller, Yuval.** *Modular Fixture - Configuration Administrator Software Requirements Specification.* [Document] s.l. : SIEMENS PLM Software, 2007.

25. **Richmond, Hans.** Pilot Environment for Modular Fixture Content Creation and Verification in Process Simulate Software. [Proposal]. s.l., Sweden : Siemens PLM Software, April 17, 2007.

26. **UGS Tecnomatix.** *eM-Planner 8.1.1 Reference Manual.* [Manual Help File] s.l. : Siemens, 2007.

27. —. *Process Designer 8.1.1 Reference Manual.* [Manual Help File] s.l. : Siemens, 2007.

28. —. *Process Simulate 8.1.1 Reference Manual.* [Manual Help File] s.l. : Siemens, 2007.

29. —. Process Designer for BIW Processing Training Guide (EMS228). 2007. MTXE228-S-081.

30. —. Process Designer Basics Training Guide (EMS127). 2007. MTXE127-S-081.

31. —. Process Simulate Basic Student Guide (WKP100S). 2007. MTXW100S-081.

32. —. *Tecnomatix Administration Guide 8.1.1.* [Manual Help File] s.l. : Siemens, 2007.

33. *Setup and Fixture planning in automated process planning.* **JONEJA, AJAY and CHANG, TIEN-CHIEN.** 7, s.l. : Taylor & Francis, 1999, IIE Transactions, Vol. 31, pp. 653-665. ISSN: 1545-8830.

34. *A Novel Modular Fixture Design and Assembly System.* **Peng Gaoliang, Liu Wenjian.** Beijing, China : IEEE, 2006. International Conference on Intelligent Robots and Systems.

35. **Wikipedia.** Modular Design. *Wikipedia.* [Online] Wikimedia Foundation, Inc. [Cited: March 18, 2008.] http://en.wikipedia.org/wiki/Modular_design.

36. **PC Magazine.** PLM Definition. *PCMag.com.* [Online] The Computer Language Company Inc., 2008. [Cited: June 6, 2008.] http://www.pcmag.com/encyclopedia_term/0,2542,t=PLM&i=49382,00.asp.

37. Overview/Functions: RobCAD. [Online] Digial Process LTD. [Cited: June 28, 2008.] http://www.dipro.co.jp/english/products/manufacturing/robcad/overview/index.html.

Appendix A) Customization Summary

As seen in chapter 4, several tools and functions have been used in order to provide a new fixture design methodology in the Process Simulate. To gain the maximum efficiency from these potentials, some customizations in the application modules are essential. This chapter will provide a compact reference to all the customizations that should be applied to Process Simulate for implementing the new design approach. Two different scopes have been defined to apply the changes: **Preparation Level** and **Administrator Level**. Several help manuals and training courses was studied and referenced in these sections (26) (27) (28) (29) (30) (31).

a) Preparation Level

Preparation level refers to all the changes that should be applied to the base structures. This includes providing new types and fields that should be filled in, later in the next level. These changes will be done one time by system supervisors and later will be used by the administrator level users.

i) CAD Application

- **Attributes Fields**

As seen before the design approach needs some attributes that the search inquiries will be based on them later. These attributes include all the possible useful design parameters that can help the fixture engineer to find the desired components/units easier. However creating and setting these attributes can be done independently in Process Simulate or Process Designer, it is strongly recommended that to provide this attributes from the CAD application and import them to the eM-Applications from there. This will help the system to keep the compatibility of the design. Example of these attributes is: part name in the database, part number, standard, design parameters such as height, length, component type and etc.

Whatever the CAD application is (NX, CATIA ...) the data will be transferred to eM-Applications via Collaborative Context in the JT format and JT format supports the attribute functionality inside itself.

ii) Process Planner

- **New Soft Classes**

There are different object types in the server database that each of them called a 'class'. These classes form a tree hierarchy that defines some parent classes and some child classes. Each child class inherits all the specification of his parent and by default called a soft class. Two major changes should be applied to soft classes using Process Planner application. One is adding the new fixture base component types **(Risers, Clamps, Brackets, Spacers ...)** under the **'Fixture'** soft class. Second one is adding the new **'Compound Fixture Unit'** soft class under the **'compound resource'** soft class. The first change adds fixture library different prototypes to the database and second change provide nodes to build a unit from a collection of fixture parts. So each unit is a 'compound fixture unit' resource that consists of several instances of different 'Fixture' part prototypes. Appendix A) shows these tree structures and how to apply them to the database.

- **New attributes for soft classes**

After adding the new object types (soft classes) to the database, the next step is to add desired attributes to each of these new soft classes. Each class can have its own special attributes to be classified with. However they can be completely new attributes, in most cases these attributes are the same as the ones which will be imported from the CAD application and pre-valued there. If an attribute is common between all the fixture objects (such as Database Name or Part Number), it is recommended to add it to the 'Fixture' soft class which is the parent of all the fixture objects. By doing this all the fixture objects will inherit these attributes from their parent class.

- **New Icons**

Each of the new soft classes can have their own specific icon in order to get better recognized. The icons should be a 16 by 16 pixel bitmap which is saved in the folder:

"...\ProgramFiles\Tecnomatix\Tecnomatix 8.1.1\InitData\DefaultCust"

NOTE 1: before applying any changes to the database through Process Planner, the eM-Server should be shut down and all open threads should be killed. This process explained in Appendix B).

iii) Customizable Tab Designer

- **Fixture Properties Tab**

For better organizing the fixture related attributes, a new tab was designed in the properties window using Tecnomatix 'Customizable Tab Designer' application. Using the Tab Designer, several related attributes can be gathered, showed and modified in a single tab of the properties window (Figure 20).

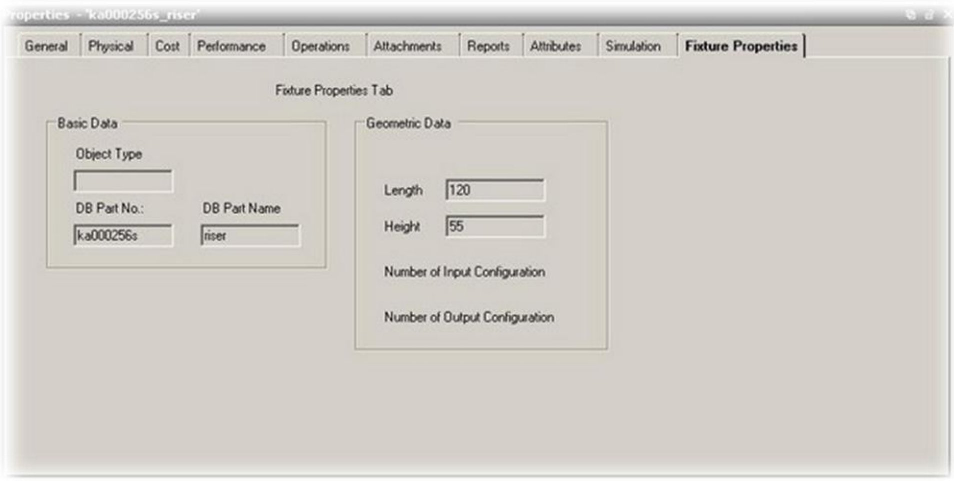


Figure 20. Properties Tab Window

Layout design of the tab is very flexible and easy. In addition, different controls such as text-box, drop down menus, different data types and etc. can be applied to the design and any attribute can be easily connected to them. Once the new tab created, the attribute values are directly modifiable from inside the new tab and so provide a very user friendly environment.

iv) Process Designer

- **Power-Bar search tool**

As seen before Power-Bar is an essential tool to search inside the library in order to find proper part or unit by using attributes as search terms. However, by default only certain objects and certain attributes has been defined in the power-bar. So prior to use, new object types, their search scope and their searchable attributes should be added to the power-bar search capabilities. This can be done by means of modifying some core files. The file is called **'SearchTool.xml'** and is located in:

'c:\documennts and settings\all users\application data\tecnomatix\process designer\8.1\'

Figure 21 shows the structure of the file and how it should be changed.

Figure 21. Powerbar Searchtool.xml file customization

- **Customizing 'New' dialog box**

'New' dialog box appears whenever the user right clicks in the navigation tree area in order to create a new object under selected branch. System administrators can determine which objects will be listed in the 'New' dialog in order to provide users with a more straightforward and less error-prone planning process. This can be applied to prevent the user to create any other data type than fixture types under the 'compound fixture units' node. The function of the 'new' command can be controlled via an XML file located at the following address:

%ClientSystemRoot%\General\NewCommandConfiguration.xml

This file determines which objects allowed to be appearing in the 'new' dialog box for each parent data type (Figure 22).

```
  </All>
- <User name="administrator">
  - <Class name="Collection">
      <PermittedChild name="CompoundOperation"
      <PermittedChild name="CompoundPart" />
    </Class>
  </User>
- <Group name="LibraryManagers">
  - <Class name="Collection">
      <PermittedChild name="OperationList" />
    </Class>
  - <Class name="CompoundResource">
      <PermittedChild name="BLABLABLA" />
    </Class>
  </Group>
- <Group name="administrators">
  - <Class name="Collection">
      <PermittedChild name="class PmResourceLibr
      <PermittedChild name="class PartLibrary" />
      <PermittedChild name="PmOperationLibrary" .
```

Figure 22. New command control file

b) Administrator Level

Administrator level customization refers to all actions should be done after Preparation Level to fill, add, build and modify the fixture related information. After the structures created in the 'Preparation Level', the administrator fills these structures in Administrator level and adding the necessary features to the fixture parts.

i) CAD Application

- **Setting the Attribute Values**

The administrator set the attribute values for each fixture object. The attribute fields created in the previous level.

- **Connecting Frames**

As mentioned before, each part should have some connecting frames to show where and how it should be connected to the other objects. These connecting frames can be created in Process Simulate or some guiding lines can be created instead of real frame in the CAD application to show the location and orientation of the connecting points.

ii) Process Simulate

- **Kinematics**

Kinematics functionalities should be added to the fixture parts in Process Simulate by administrator. Administrator should check out the kinematic needed objects, enter the modeling mode, add the kinematic chains, and save them back to the library.

- **Signals**

Signals also should be added to the desired fixture parts in Process Simulate. Signals will be added to actuating objects such as clamps and to sensing objects such as switches. In addition, some control units exist in each fixture, which will control the entire fixture. Some logic blocks should be added to these units to make them ready for use.

iii) Process Designer

- **Filling And Importing The Fixture Library**

Fixture Part Library, Fixture Template Library, and Fixture Archive Library are imported and getting filled by the administrator. As the structures created before and ready to use, in this level the administrator just filling the structures and arranging them.

Appendix B) eM-Server Shutdown Precautions

Before applying any change to the database, the eM-Server should be shutdown using the following procedure in the **'Admin Console'** application (it can be found in Tecnomatix programs start menu). This action has five steps:

1. In order to ensure that no user will login to the system while administrative tasks are being performed, use the **eMServer Administration->Enable/Disable Login** tab. Click to disable all logins from the users while the changes are applied.

2. Kill all the active sessions by first asking all the users to log out the system. Then from the **'AdminConsole'** window, select the **eMServer Administration->Session Administration node.** Select all the open sessions in the window and press 'kill the sessions' button after getting sure that no user is logged in.

3. Click **Shutdown**, in order to shut down the eMServer in the current machine by using the **eMServer Administration->eMServer Shutdown** tab.

4. Now apply all the necessary changes to the database using '**eM-Planner**' application.

5. Enable the login process again for the users by doing the same step mentioned in number 1.

For further information about this procedure, refer to the Tecnomatix Administration Guide (32).