



Manufacturing traceability automation using features and nc-functions.

Julio Garrido Campos, Martin Hardwick

► To cite this version:

Julio Garrido Campos, Martin Hardwick. Manufacturing traceability automation using features and nc-functions.. International Journal of Computer Integrated Manufacturing, 2009, 22 (02), pp.112-128. 10.1080/09511920802132714 . hal-00513401

HAL Id: hal-00513401

<https://hal.science/hal-00513401>

Submitted on 1 Sep 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Manufacturing traceability automation using features and nc-functions.

Journal:	<i>International Journal of Computer Integrated Manufacturing</i>
Manuscript ID:	TCIM-2007-IJCIM-0090.R1
Manuscript Type:	Original Manuscript
Date Submitted by the Author:	29-Nov-2007
Complete List of Authors:	Garrido Campos, Julio; Universidad de Vigo, Departamento de Ingenieria de Sistemas y Automatica Hardwick, Martin; RPI, Department of Computer Science
Keywords:	STEP NC, STEP, CNC, COLLABORATIVE MANUFACTURING
Keywords (user):	Traceability



1. Introduction

1.1. New Information Requirements for Collaborative Manufacturing Environments: Traceability

Manufacturing traceability can be defined as a set of practices to make available in the future all essential information about (Simchi-Levi et al. 2003) how a product has been manufactured. The objective of manufacturing traceability is to provide the necessary knowledge about the manufacturing process and set up mechanisms to react against defects or wrong behaviours shown in final products, but originating during the manufacturing phase (Cheng and Simmons 1994, ECR 2004, ISO 2000).

Rapid advances in information technologies are causing companies to adopt new manufacturing models that integrate CAD (computer aided design), CAM (computer aided manufacturing) and CNC (computerized numerical controllers) models into collaborative environments (Suh et al. 2002). In this paper we show how manufacturing traceability processes can be implemented within these environments using the STEP standards (ISO 10303-1 1994). We show how these standards can be extended to enable the integration of real-time shop information, and collect manufacturing data in real-time.

1.2. Traceability Integration and Automation in CAD/CAM/CNC chains

Research in collaborative CAD/CAM/CNC has traditionally focused on finding solutions for CAD/CAM and CNC data exchange and integration (Xu and He 2004). The issues addressed include the inability to interrelate product data, disconnected product information, visibility across the supply chain, interoperability, quality of data and data integrity.

As CAD/CAM/CNC manufacturing systems and processes are becoming increasingly complex, better information gathering and analysis techniques are needed to enable the future development of intelligent manufacturing systems. Requirements for these systems include process monitoring and manufacturing traceability. The first is necessary to supply data to develop dynamic and global optimization tools and methodologies. The second is necessary to support off-line quality control, long-term manufacturing data storage, etc.

The main problem for upstream integration of shop-floor data has been to couple machine and process observations with actual executed machining workingsteps, (Brecher et al., 2006). STEP-NC, provides higher information to the CNC, making shop-floor data more practical because it can be explained using the high-level concepts in the new language. The approaches for process data feed back in STEP-NC that have been described in the literature include descriptions of extended STEP data models for shop-floor data representation (Garrido and Hardwick 2006, Wosnik et al. 2006), and proposals for implementing data acquisition systems with embedded STEP-NC information (Fichtner et al. 2006).

This paper describes a proposal to integrate and automate manufacturing traceability (information requirements and processes) in a STEP-enabled manufacturing

environment. A feature-based approach is used to integrate traceability information with other CAD product and CAM process information. The paper shows how the feature-based approach enables downstream to upstream traceability automation. The proposal takes advantage of the new STEP-NC programming language to map CAD/CAM level traceability information requirements into directly interpretable machining tasks. The paper shows how the new approach simplifies traceability integration in emerging DABA (design anywhere build anywhere) manufacturing systems (Chen et al. 2003, Li et al. 2005).

The paper is organized as follows. Section 2 presents the STEP-Manufacturing environments and STEP-NC. Section 3 describes the proposed traceability nc_functions information model. Section 4 presents a prototype implementation based on this model. Finally, Section 5 contains our conclusions and proposals for future work.

2. STEP standard and STEP-Enable Manufacturing Environments

2.1. STEP Manufacturing Environments: STEP-NC

STEP (ISO 10303: S**Standard for the Exchange of Product model data**) is widely used as a technology for the file-transfer of design data. For example AP-203 (application protocol) is used to exchange design data among CAD systems and between CAD and CAM systems (ISO 10303-203 1994). New advances in CAD/CAM/CNC collaborative systems are promoting the development of new APs to support PDM (product data management) technologies, real-time communication and collaboration during different phases of product development.

AP **product** data is organized into UoF (units of functionality), covering specific information requirements for different activities. When these information requirements span several product development stages, they can be harmonized by developing a common UoF data model that is included in several AP's. An example is the feature UoF (shape design and process features) used by the "SMS" APs (the STEP Manufacturing Suite of application protocols)(OMAC 2005).

Figure 1 shows a manufacturing scenario for mechanical parts and assemblies based on design features using the SMS APs. In this case, the "STEP in, STEP **out**, STEP throughout" data flow from design to **manufacturing allows** a direct exchange and a better information connection between CAD/CAM and NC systems.

The "primary data" workflow for Figure 1 is as follows:

- In the design phase, the product is specified using the AP-203 Ed2 product model that includes geometric and tolerance data. AP-203 also includes basic PDM data so it is suitable for CAD data exchange. However it does not include the feature information needed for CAM level planning of the product machining.
- In the detailing phase, the AP-203 information is processed by a detailing system and converted to AP-224 (ISO 10303-224 2001) data containing the original AP-203 geometric information and new information describing the features that need to be machined and their associated tolerances.

- In the process planning phase, macro process planning information about facilities, numerical controls available, and the distribution of work to each manufacturing node is added to the AP-224 information to create AP-240 (ISO 10303-240 2005) planning information.
- In the manufacturing phase, shop-floor or micro process planning data for programming CNC controllers is added to the AP-224/240 information to create STEP-NC AP-238 (ISO 10303-238 2007) feature machining information.

Features are the kernel technology of CAD/CAM integration (Wang and Zhang 2002) and they are the common denominator between all the APs (except AP-203) in the Figure 1 environment.

2.2. Into STEP-NC executable structure

STEP-NC (ISO 14649, ISO 10303-238) is a high level programming, with a richer interface than ISO 6983 (ISO 6983-1 1982, Xu 2006, Xu and Newman 2006), and an object oriented data model, for transferring information between CAD/CAM programming systems and computerised numerical controllers (CNC). It allows a programmer to describe not only “how to make” the piece, but also “what to make”, by incorporating feature-based product knowledge right into the CNC controller programs, as part of the information requirements available for NC programming.

The STEP-NC AP-238 information requirements are organized into UoFs covering geometric information (workpiece, geometric tolerances, tool path, manufacturing features...), control structures for the program execution (executable), technological information (process data for milling, tools for milling...) and data management. AP-238 programs can be described as a set of complex and structured tasks (workingsteps) to machine the features of a workpiece. The “executable” UoF specifies the information requirements to describe the control flow of the machining program, as well as some non-machining actions that may be performed by NCs. Its base entity is the “executable” entity from which other executables objects describing actions that can be started or completed on a machine are derived. These derived entities can be classified into three types of objects:

1. **Workingsteps** describe manufacturing or handling operations which involve axis interpolation. They must be performed over a workpiece and they are the essential building blocks for NC programs.
2. **NC functions** describe switching operations or other non-interpolating machine functionality such as turning the coolant on.
3. **Program structures** are used to build logical blocks for programming the manufacturing operations. These program structures have the authority over the actual manufacturing sequence. The most important is the **workplan**, which combines several workingsteps, NC functions and other (nested) workplans into a sequential order.

Figure 2 shows an EXPRES-G diagram for the STEP-NC executable. As shown there are several types of program structures for arranging different types of manufacturing sequences. The diagram also lists the nc_functions included in the first edition of STEP-NC.

3. STEP-NC: Traceability activities automation: nc_functions

3.1. New Information requirements in STEP-enable manufacturing scenarios: Feature-based Traceability Processes

Taking advantage of the feature-based harmonized STEP data flow from Figure 1, this paper addresses the integration and automation of manufacturing traceability information and processes in a CAD/CAM/CNC supply chain. Figure 3 identifies – in a high-level AAM (application activity model) - the main manufacturing traceability tasks - configuration, data **collection and data reporting**.

The implementation of these activities results in an “extended data workflow” characterized by a bidirectional data flow between CAD/CAM enterprise level systems and shop-floor level CNC systems. Through the downstream data flow, traceability configuration requirements are communicated from qualified data sources to the shop-floor (steps 2 and 3 in Figure 4). These requirements are interpreted on the shop-floor and result in automated data collection (step 4 in Figure 4). The data can then be communicated back by following an inverse upstream data path to the configuration levels (step 5 in Figure 4).

In the proposed feature-based traceability approach shown in Figure 4, traceability configuration data (information requirements) are linked to shape features during the CAD/CAM design phase. Feature data is processed by CAM systems to create feature-based process data. Traceability configuration information requirements are specified at the CAM level to create workingstep-based traceability tasks to be done during the CNC machining. The knowledge about product data features made available to the CNC is the key to this approach. Feature or workingstep based data collected during the machining phase is communicated back to the CAM/CAD systems to do data management or other advanced closed loop tasks.

3.2. STEP-compliant Traceability NC_functions

Traceability automation is achieved by translating feature-based traceability requirements into workingstep-based traceability tasks or operations. The proposed method to do this in AP-238 programs is to parameterize these traceability tasks as new AP-238 “nc_function” entities.

As explained in Section 2.2, AP-238 nc_functions are machining operations completed by the NC-controller but which do not imply machine axis movements. Figure 5 shows how the proposed traceability nc_functions are incorporated into the STEP-NC model. As will be shown in Section 4.1, adding the appropriate nc_function in the AP-238 executable structure before (start an action) and after the corresponding workingsteps (to stop the corresponding action) makes possible the implementation of feature-based traceability tasks.

The new nc_functions have been divided into three groups according to how they interact with the rest of the machining program.

The functions in the first group are used to get single values (non concurrent data collection). As shown in the left part of Figure 6 these functions have been modelled to be blocking or modal functions. They are inserted in the NC program, and they are executed as part of the machining NC program execution flow. The machining program does not continue to perform the next action (executable element) until a value has been returned and stored in the corresponding *nc_variable*. These values can be tested at run time or by an application that reads them after run time. Figure 7 is the EXPRESS-G specification for all traceability *nc_functions*, and Group I functions are in the left part of the figure.

Group II functions define traceability actions that must be performed by the controller concurrently with other CNC machining operations (Figure 6 centre). Group II functions are divided into pairs of start and stop process functions. When a start function is found by the NC Controller, it activates the corresponding concurrent data monitoring process. This process remains active until the counterpart stop function is found in the program.

Group II functions monitor the execution of tool paths and allow an application to check things afterwards. Therefore, they only apply to actions that have a tool path. These functions put their results into a data structure compatible with the bounded curve of a tool path so they can be analyzed off-line by an application that will read the bounded curve sensor data and make comparisons with the corresponding tool path. This data structure is the *results_data* entity in Figure 7. It has two attributes: “*measured_for*” which is a link to the measured tool path and “*measured_results*” is a *bounded_curve*.

In order to relate the data, the results bounded curve is required to have the same parametrization as the tool path curve, meaning they must start and stop their composite curve segments at the same points. The type of the curve segments does not have to be the same. For example, if the path segment is an arc, the *measured_results* can be a one dimensional polyline segment with the end point storing the as-measured maximum deviation of the monitored variable as the cutting tool moves along the arc.

Group III functions are the same as Group II but with the extended ability to perform actions if a threshold is crossed (right part in Figure 6). The purpose is to be able to react to “in-process” collected data values. These actions are specified in the *its_action* attribute, which is a workplan type. Each one of the functions in the right part of Figure 7 may behave as Group II or Group III. If a workplan and a threshold value are specified, then the Group II function becomes a Group III. Finally, to support Group III functions, a *set_alarm* function is defined to enrich the range of actions that can be taken.

Table 1 describes the proposed *nc_functions* in more detail and Section 4 describes how these functions have been implemented and tested.

4. Implementation Testing

The application scenario shown in Figure 4 (section 3) has been implemented to demonstrate and validate the proposed AP-238 *nc_functions*. CAT (Computer Aided Traceability) tools have been developed based on the STEP Tools STEP-NC dll

explorer, to perform the data configuration, machining simulation and analysis of the collected data.

A STEP-NC native CNC simulator and a tool path explorer has been coded in C++ under MS Visual Studio 2005, using ROSE STEP C++ libraries provided by STEP Tools ST-Developer v 11.0 (STEP Tools 2007), Open Cascade graphic libraries (Open CASCADE 2004) and MS XML API. The System prototypes have been tested with the OS of Windows XP and .NET.

The simulator also includes a post-processor module to translate some of the proposed STEP-NC nc_functions into proprietary Siemens Sinumerik 840d code, enabling the test of these data collection tasks in G & M based CNC controllers.

4.1. Setting up information requirements: traceability and monitoring configuration

As described in Section 3, traceability configuration is the first task to address. Configuration means mapping data collection requirements into machine understandable information specifying which data must be collected during shop-floor machining (semantic data scope), and indicating when to start and finish collecting the data (temporal data scope). Both objectives are covered for STEP CNC machining environments with the proposed nc_function approach.

Figure 8 describes the activity-based architecture of the implemented application configuration module. This module allows the user to select the information requirements to be set over different features, workingsteps or even tool paths, and integrates these information requirements by translating them into the proper nc_functions to generate a “traceability extended” AP-238 file ready to be interpreted by the shop-floor NC machines. This module has two main parts: the “Data Mapper” and the “Function Parser”.

The “Data Mapper” (A11 in Figure 8) extracts the essential data for the configuration application. It reads the AP-238 executable structure and tool path geometry to make a feature-based tree view of the workpiece, creating child nodes for the workingsteps, operations and associated tool paths. Figure 9 shows how the application processes the NAS 979 part data (widely used as a STEP-NC test piece). In this case, the workpiece is described as a freeform feature to be machined, so the sequence of workingsteps for this feature is shown. In the demonstration example, workingstep execution time, and maximum path deviation positions for the composite curve tool path 3 (AP-238 entity #3066 in Figure 9) of the second workpiece workingstep (AP-238 entity #10794 in Figure 9) are configured as requirements to be monitored.

After selecting the traceability requirements, the “AP-238 Configuration” (A12 in Figure 8) module is ready to translate the selected user requirements into AP-238 compliant traceability nc_functions. This implies not only inserting Group I or Group II functions and data placeholders at the right places in the executable structure but also, as shown in Figure 10, splitting a workingstep into several workingsteps to allow specific tool path monitoring.

In Figure 10, the left tree corresponds to the un-parsed AP-238 executable structure, and the right tree represents the parsed AP-238 executable structure, where two `get_time` (AP-238 entities #20000 and #50000 in Figure 10) functions have been inserted to measure the total workingstep time. This workingstep has also been split in three parts to allow the insertion of the corresponding Group II start (AP-238 entity #30000 in Figure 10) and stop monitoring (AP-238 entity #40000 in Figure 10) functions before and after the selected tool path execution.

4.2. Data collection: traceability execution

At the shop floor, a CNC simulation module (Figure 11), reads the extended AP-238 file to simulate the execution of the machining program in traceability mode.

The A21 module interprets the extended AP-238 file, but in order to implement the proposed traceability `nc_functions` (ARM model of Section 3, Figure 7) it is necessary to map the ARM model into and AIM model using STEP generic resources. Each `nc_function` is mapped into the “machining_nc_function” type, a special subtype of the “machining_process_executable” following the model used for AP-238 `nc_functions`. Specific `nc_functions` are identified for the application by reading its “description” constrained string attribute. For example, a string value of “load_tool” identifies the `nc_function` as the `load_tool` function.

For Group I `nc_functions`, the returned value has been modelled using the `nc_variable` type, as shown in the sample code below (XML part 28), for one of the `get_time` `nc_functions` configured in previous section.

```
<!-- ***** <br>
* Application object: GET_TIME (id20000)
* ITS_TIME: id20001, id20002, id20003, id20004, id20005, ['GET TIME]-
*-->

<Machining_nc_function id="id20000" Name="line 7" Description="get time" Consequence="Record Time" Purpose="Get WS
Start Time"/>
<Action_property id="id20001" Name="expected value" Description="get_time function">
  <Definition>
    <Machining_nc_function ref="id20001" xsi:nil="true"/>
  </Definition>
</Action_property>
<Action_property_representation id="id20002" Name="" Description="get_time function " Property="id20001"
Representation="id20003"/>
<Representation id="id20003" Name="" Items="id20005" Context_of_items="id1000"/>

<!-- ***** <br>
* Application object: REAL_VARIABLE (id20005)
* ITS_NAME: id20005, ['Start Time']
* INITIAL_VALUE: id20005, id20006, id20007, [null]
*-->
<exp:complexEntity id="id20005" entities="expression_representation_item real_numeric_variable">
  <Representation_item Name="Start Time"/>
  <Generic_expression/>
  <Expression_representation_item/>
  <Expression/>
  <Numeric_expression/>
  <Simple_generic_expression/>
  <Simple_numeric_expression/>
  <Generic_variable/>
  <Variable/>
  <Numeric_variable/>
  <Real_numeric_variable/>
</exp:complexEntity>
```

The A22 module drives the machining simulation shown in Figure 12. The A23 module is an interpolation module, which simulates the CNC tool path interpolation. These data are used by the A24 module to display the machining simulation and also by the A25 module, responsible for the traceability process.

The A25 module implements the NC functions using CNC dependant algorithms. For Group II and III functions this requires setting a number of control points over each parameterized segment of the tool path trajectory.

In the example, when the “start monitoring maximum deviation positions along tool path” nc_function is executed, the monitoring module creates control points for reading the actual tool position (machine encoder positions from module A23) so that it can estimate the tool path deviation (dashed line in Figure 13). The algorithm used calculates the projection p_i (points marked with stars in Figure 13) of the measured position q_i (points marked with circles in Figure 13) along the parametrized trajectory and computes a deviation value using the formula (1) shown below.

$$(1) \ \varepsilon_i = p_i T_{CM}^{ref} - q_i T_{CNC}^{ref} \quad i = 1, n$$

By comparing these values applying equation 1, where T_{CNC}^{ref} and T_{CM}^{ref} are homogeneous transformation matrix to transform to the same reference coordinate system, an estimation vector of the deviation error is obtained. These values are then stored to compute the maximum deviation value for each parametrized tool path segment. When the tool path is completed these values are recorded in a “bounded curve structure” (a list of maximum deviation values) with the same parametrization, in terms of start and end points, as the executed tool path.

Functions have been also tested (Figure 14) with current technology implementations. A G & M code post-processor (A27 module for the Siemens Sinumerik 840d) translates AP-238 code into specific CNC controller code (Figure 15). Using specific CNC resources, G & M codes, custom macro B programming capabilities (specific machine programming codes that extends the standard G-code to enhance part programming) or other CNC OEM programming capabilities like synchronous motion actions, the post-processor implements some of the proposed traceability nc_functions.

In this case the collected data is logged in ASCII files by including enough information (AP-238 entity identities) to link them to the corresponding AP-238 workingsteps or tool paths.

As shown in Figure 15, the time function has been implemented by using 840D write functions to log internal time variables that include information about the AP-238 identities of the associated workingsteps.

For the maximum position deviation tool path function, the tested implementation is based on siemens “synchronous motion actions”. Using these custom capabilities, an action is conditionally performed at the end of each interpolation cycle depending on an evaluated condition. For the position deviation function, the condition is the comparison between the previously computed maximum deviation and the actual cycle position set

point deviation. As a result at the end of each G motion code, the maximum deviation values and position are recorded and can be written to a log file.

4.3. Using *the* Collected Data: closed loop data management

Finally, with the collected data several tasks, including feedback for close loop decision analysis or audit, can be done. The implicit linkage between collected data and shared STEP CAD-CAM-CAPP-CNC product data simplifies these tasks. The instantiated STEP “result data” shown below, uses a bounded curve compatible structure with a reference to the measured tool path. This enables a linkage with higher AP-238 entities, like operations, workingsteps and features, and their interpretation at any CAD/CAM level sharing this STEP information.

```
<!-- ***** <br>* Application object: START_MONITORING (id30000)<br>* ITS_VALUES: id30001, id30002, id30003, id30004, id30005, ['MAX DEVS']<br>* ITS_ID: id30001,<br>* --><br><Machining_nc_function id="id30000" Name="" Description="start_monitoring_max_deviation_along_toolpath "<br>Consequence="Record Toolpath Deviatio " Purpose="Start Monitoring "/><br><Action_property id="id30001" Name="measured_for" Description="deviation values "><br>  <Definition><br>    <Machining_nc_function ref="id30000" xsi:nil="true"/><br>  </Definition><br></Action_property><br><Action_property_representation id="id30002" Name="" Description="cutter location trajectory" Property="id30001"<br>Representation="id30069"/><br><Action_property id="id30003" Name="basic curve" Description="deviation values "><br>  <Definition><br>    <Machining_nc_function ref="id30000" xsi:nil="true"/><br>  </Definition><br></Action_property><br><Action_property_representation id="id30004" Name="" Description="deviations along toolpath " Property="id30003"<br>Representation="id300045"/><br><Representation id="id30005" Name="" Items="id30006" Context_of_items=""/><br><Polyline id="id30006" Name="basic curve for deviation values" Points="id30007 id30008 id30009 id30010 id30011 id30012<br>id30013 id30014 id30015"/><br><Cartesian_point id="id30007" Name=" " Coordinates="0.007097000 0 0"/><br><Cartesian_point id="id30008" Name=" " Coordinates="0.003570000 0 0"/><br><Cartesian_point id="id30009" Name=" " Coordinates="0.013400000 0 0"/><br>.....
```

Figure 16 shows how these data can be graphically browsed over AP-238 tool path representations in the demonstration application. Maximum deviation values for each executed tool path segment, obtained from measures done at the selected control points (simulation results list in Figure 15), can be graphically browsed over the tool paths (numeric values in the graphic view).

5. Conclusions and Future work

This paper has shown how a feature-based traceability approach can be efficiently integrated with other STEP based CAD/CAM/CNC processes. With the proposed approach CNC controllers can interpret STEP-NC traceability commands, execute the required traceability and feedback the manufacturing data for the features associated with the monitored processes.

The traceability and data monitoring nc_functions presented here (Figure 7 and Table 1) are being considered by the ISO TC184/SC4 STEP-Manufacturing committee for inclusion in the second edition of STEP AP-238. The Group I, II, and III functions

presented in the paper have been tested in the implemented AP-238 simulation. Group I functions were also tested in current technology implementations by mapping requirements into M & G code functions for specific controllers. Simplified versions of Group II and III were also tested on current controllers.

Future work will expand the functions and go deeper **into** data structures definitions. For instance, there is an issue if there is no explicit tool path (e.g. if data is Conformance Classes 3 or 4: CC3 or CC4 of AP-238). In that case, the controller will decide its own tools path based on the feature information in the AP-238 file. Future work for this situation will define the standard extensions for tracing of semantic operations (e.g. mill a pocket).

Another area for future work is defining or using better data structures for easier closed loop programming, e.g. a Group I function that gets the current point as a STEP point instead of three functions to get x, y and z. This requirement is not unique to traceability however because better data structures will make closed loop programming more efficient for integrated machining and measurement applications as well.

To improve storage efficiency a way is needed to divide the data so that the results do not have to go into the same file as the AP-238 input. Again this requirement is not unique to traceability.

Finally, new traceability functions should be considered for future extensions. For example, functions to collect data about asynchronous or random events coming from the CNC (alarms, warning, operation mode changes and so on).

References.

- Brecher, C., Vittr, M. and Wolf, J., 2006. Close-loop CAPP/CAM/CNC process chain based on STEP and STEP-NC inspection tasks. *International Journal of Computer Integrated Manufacturing*, 19 (6), 570-80.
- Chen, L., Song, Z. and Feng, L., 2003, Internet-enabled real-time collaborative assembly modeling via an e-Assembly system: status and promise. *Computer-Aided Design*, 36 (9), 835-47.
- Cheng, M.J. and Simmons, J.E.L., 1994. Traceability in Manufacturing Systems. *International Journal of Operations and Production Management*, 14 (10), 4-16.
- ECR, 2004. ECR Blue Book – Using traceability in the Supply Chain to meet Consumer Safety Expectations [online]. Available from: <http://www.ecrnet.org/> [Accessed 2 August 2007]
- Fichtner, D., Nester, A., Dang, N., Scgulze A., Carlsen, U., Schreiber, S. and Klemm, P., 2006. Use of agents and neural networks for adquisition and preparation of distributed NC information to support NC planning. *International Journal of Computer Integrated Manufacturing*, 19 (6), 581-92.
- Garrido J. and Hardwick, M. 2006. A traceability information model for CNC manufacturing. *Computer-Aided Design*, 38 (5), 540-51.
- International Organization for Standardization, ISO 6983-1, 1982. Numerical Control of Machines — Program Format and Definition of Address Words – Part 1: Data format for positioning, line motion and contouring control systems, ISO TC 184/SC 1, Geneva, Switzerland.

- International Organization for Standardization, ISO 10303-1, 1994. Industrial automation systems and integration - Product data representation and exchange – Part 1: overview and fundamentals principles, ISO TC 184/SC 4, Geneva, Switzerland.
- International Organization for Standardization, ISO 10303-203, 1994. Industrial automation systems and integration - Product data representation and exchange – Part 203: configuration controlled design, ISO TC 184/SC 4, Geneva, Switzerland.
- International Organization for Standardization, ISO 9001, 2000. Quality Management Systems – Requirements, ISO TC 176/SC 2, Geneva, Switzerland.
- International Organization for Standardization, ISO 10303-224, 2001. Industrial automation systems and integration - Product data representation and exchange - Part 224: mechanical product definition for process planning using machining features, ISO TC 184/SC 4, Geneva, Switzerland.
- International Organization for Standardization, ISO/CD TS 10303-240, 2005. Industrial automation systems and integration - Product data representation and exchange – Part 240: Application protocol: Process plans for machined products, ISO TC 184/SC 4, Geneva, Switzerland.
- International Organization for Standardization, ISO 10303-238, 2007. Industrial automation systems and integration - Product data representation and exchange – Part 238: application interpreted model for computerized numerical controllers, ISO TC 184/SC 4, Geneva, Switzerland.
- Li, W.D., Lu, W.F., Fuh, J.Y.H. and Wong, Y.S., 2005. Collaborative computer-aided design - research and development status. *Computer-Aided Design*, 37 (9), 931-40.
- OMAC, 2005. STEP-NC Pilot Demonstration. OMAC STEP-NC Working Group Meeting [Online]. Available from: www.isd.mel.nist.gov/projects/stepnc/omac-orlando-2005/pilot-scenario.pdf. [Accessed February 2007]
- Open CASCADE, 2004. Technical overview 4/2002 [online]. Available from: www.opencascade.org. [Accessed 1 July 2007]
- Simchi-Levi, D. Kaminsky, P., & Simchi-Levi, E., 2003. *Designing and management the supply chain*. McGraw-Hill Irwin, New York, NY.
- StepTools, 2007. ST-Developer on line manuals [online]. Available from: http://www.steptools.com/Support/stdev_docs/. [Accessed 2 August 2007]
- Suh, S.H., Cho, J.H. and Hong, H.D., 2002. On the architecture of intelligent STEP-compliant CNC. *International Journal of Computer Integrated Manufacturing*, 15 (2), 168-177.
- Wang, H-F. and Zhang, Y.L., 2002. CAD/CAM integrated system in collaborative development environment. *Robotics and Computer-Integrated Manufacturing*, 18 (2), 135-45.
- Wosnik, M., Kramer, C., Selig, A. and Klemm, P., 2006. STEP-NC for integrated and Distributed Manufacturing Processes: Enabling feedback of process data by use of STEP-NC. *International Journal of Computer Integrated Manufacturing*, 19 (6), 559-70.
- Xu, X.W. and He, Q., 2004. Striving for a total integration of CAD, CAPP, CAM and CNC. *Robotics and Computer-Integrated Manufacturing*, 20(2), 101-9.
- Xu, X.W. and Newman, S.T., 2006. Making CNC machine tools more open, interoperable and intelligent - a review of the technologies. *Computers in Industry*, 57 (2), 141-52.

Xu, X.W., 2006. Realization of STEP-NC enabled machining. *Robotics and Computer-Integrated Manufacturing*, 22 (2), 144-53.

For Peer Review Only

Figure 1. STEP-enable scenario for manufacturing of mechanical pieces.

Figure 2. The AP-238 executable UoF ARM.

Figure 3. Traceability application activity model.

Figure 4. Traceability activities scope and extended data workflow.

Figure 5. Modularity of existing AP-238 ARM nc_functions and the traceability proposal.

Figure 6. Traceability nc_functions and concurrent process execution.

Figure 7. Sample STEP-NC compliant traceability nc_functions.

Figure 8. IDEF0 activity diagrams for the implemented traceability and monitoring configuration tool.

Figure 9. Configuring traceability data requirements over AP-238 data.

Figure 10. AP-238 executable parsing process.

Figure 11. IDEF0 activity diagrams for the implemented traceability and monitoring simulation tool.

Figure 12. AP-238 CNC simulation.

Figure 13. Maximum tool path deviation evaluation.

Figure 14. G&M monitoring post-processed sample file.

Figure 15. G&M CNC simulation.

Figure 16. Traceability report.

Function group	Function name	Purpose
I	<i>get_time</i>	It gets the present time in the controller.
	<i>get_operator_id</i>	It gets the identification of the machine operator.
	<i>get_machine_id</i>	It gets the identification (unique identification: serial number) of the machine being used.
	<i>get_tool_id</i>	It gets the identification (unique identification or lot identification) of the cutting tool in use.
	<i>get_coolant_id</i>	Gets the identification of the coolant in use.
	<i>get_raw_material_id</i>	It gets the identification number (or the lot number) of the raw workpiece.
	<i>get_x_location_data</i>	It gets the actual value of the x axis. (the same for y, z, i, j, k, axis)
	<i>get_sensor_data</i>	It gets the actual value of a sensor. It is a parametric function: the specific sensor is identified by the <i>sensor_id</i> attribute value.
II/III	<i>start_measuring_maximun_deviation_position_along_toolpath</i>	It starts the measurement of the maximum deviation of the tool position along the tool path. For each tool path segment, the function measures the maximum deviation of the actual position of the tool from the calculated one. A unique deviation value, the maximum, is saved for each segment. (the same for: axis: tool axis federate: tool feed rate spindle_speed: tool spindle speed)
	<i>stop_measuring_maximun_deviation_position_along_toolpath</i>	It stops the process
	<i>start_monitoring_sensor_data_along_toolpath</i>	Parametric function. It starts the monitoring of a sensor value identified by the "sensor_id" attribute value.
	<i>stop_monitoring_sensor_data_along_toolpath</i>	It stops the monitoring of the sensor identified by the "sensor_id" attribute value.
III (special)	<i>set_alarm</i>	Specifies an alarm number to be communicated to the controller.

Table 1. Traceability nc_functions.

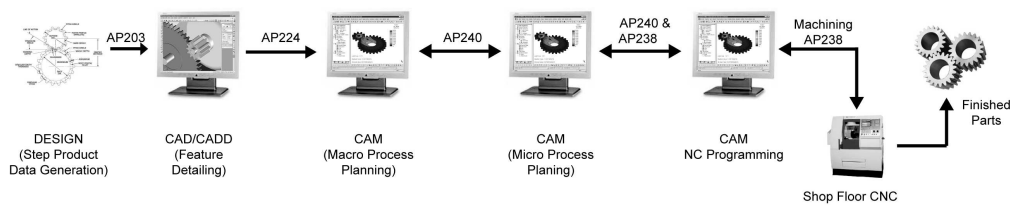


Figure 1. STEP-enable scenario for manufacturing of mechanical pieces.
219x44mm (300 x 300 DPI)

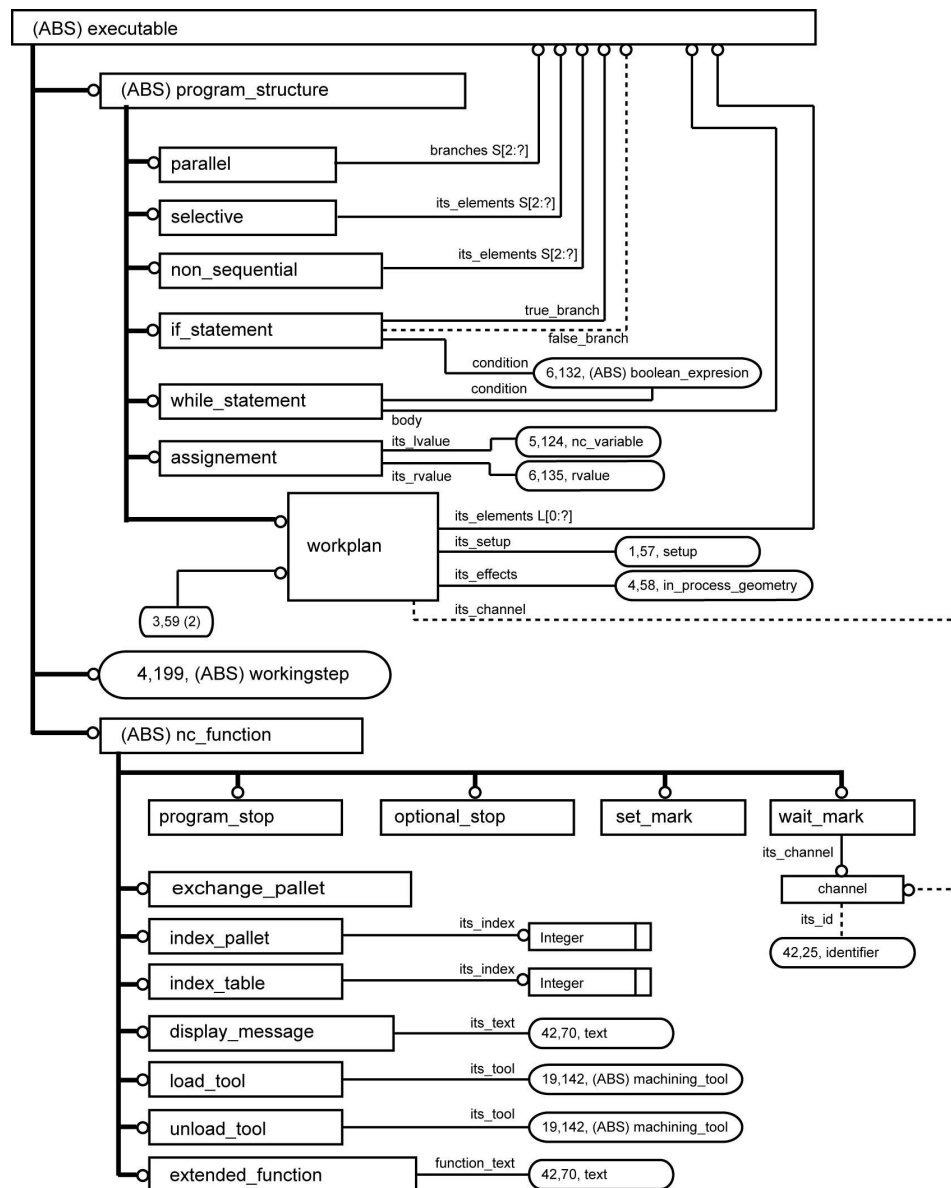


Figure 2. The AP-238 executable UoF ARM.
143x181mm (300 x 300 DPI)

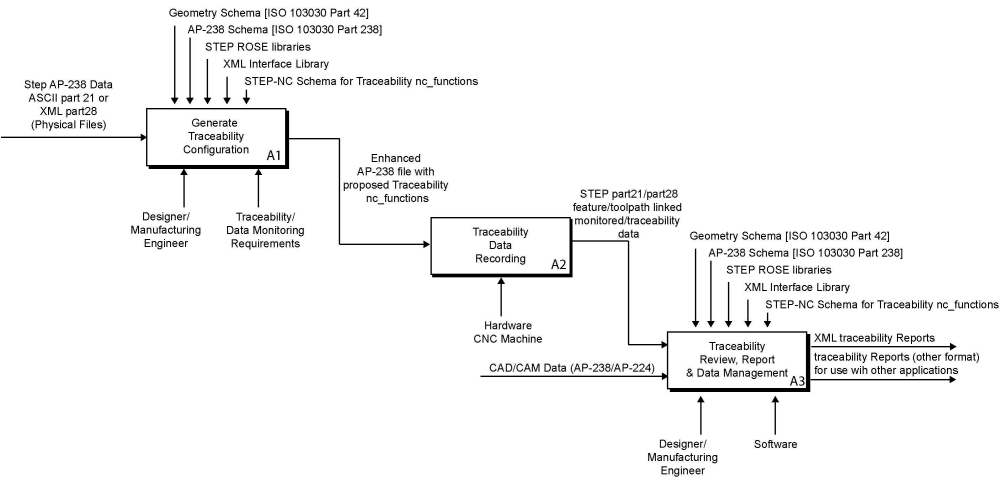


Figure 3. Traceability application activity model.
254x119mm (300 x 300 DPI)

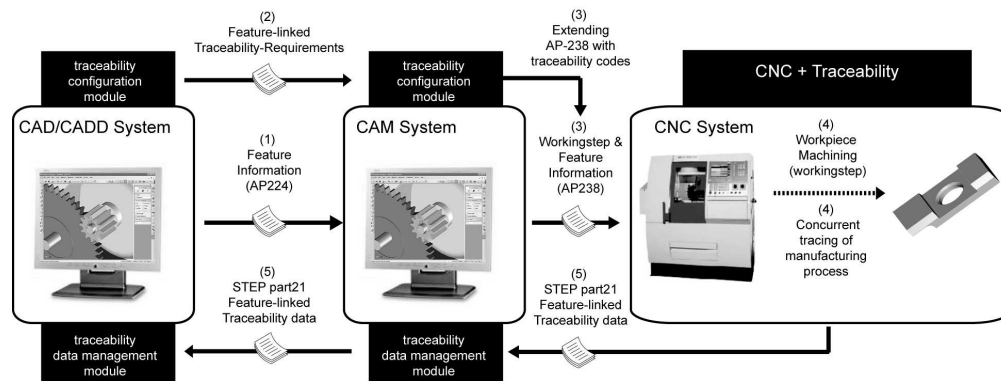


Figure 4. Traceability activities scope and extended data workflow.
164x77mm (300 x 300 DPI)

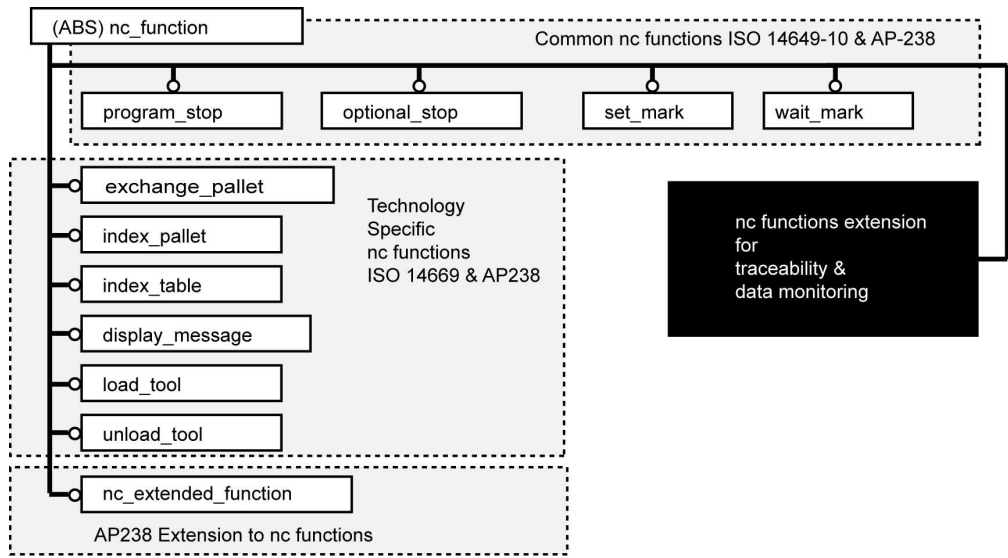


Figure 5. Modularity of existing AP-238 ARM nc_functions and the traceability proposal.
145x80mm (300 x 300 DPI)

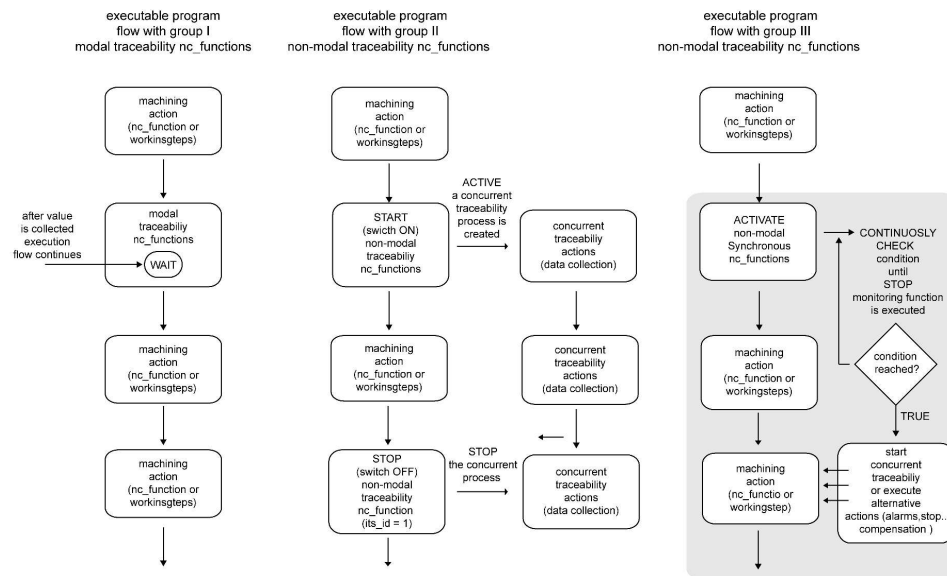


Figure 6. Traceability nc_functions and concurrent process execution.
328x187mm (300 x 300 DPI)

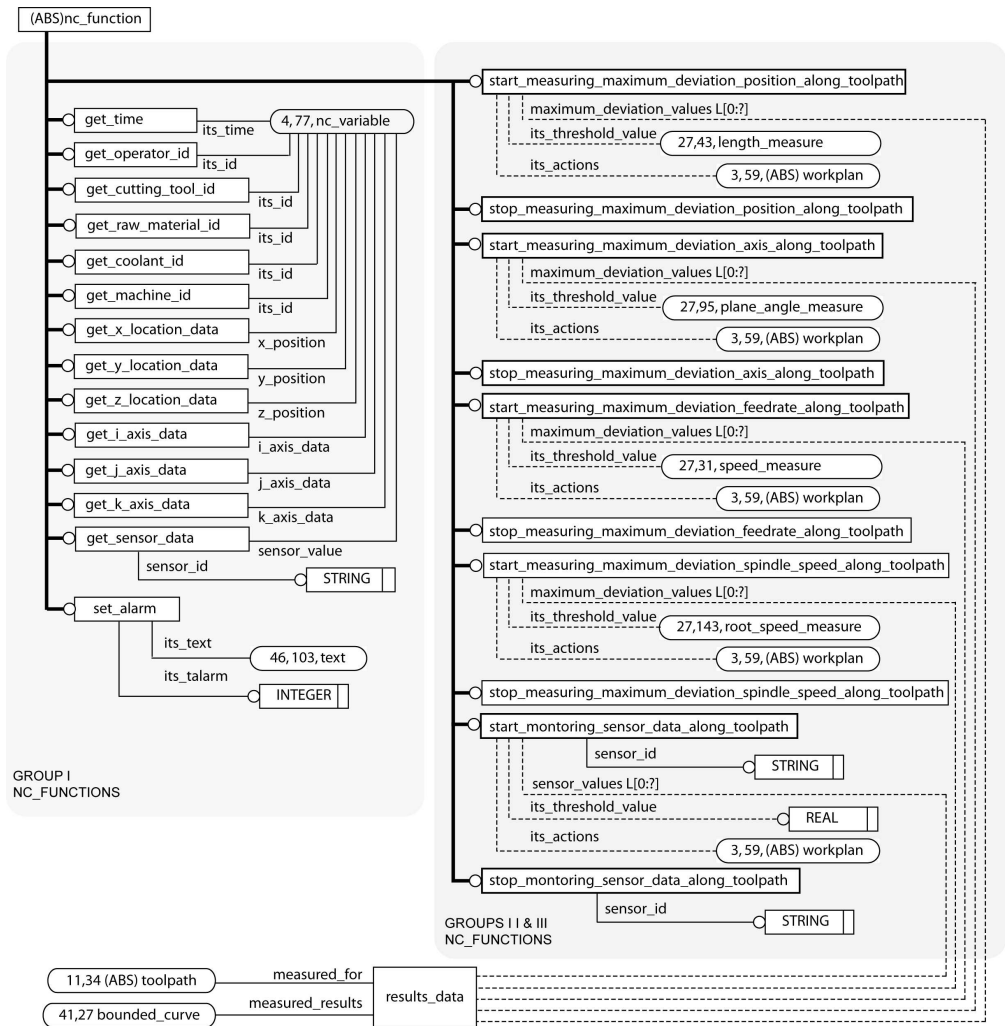


Figure 7. Sample STEP-NC compliant traceability nc_functions.
206x210mm (300 x 300 DPI)

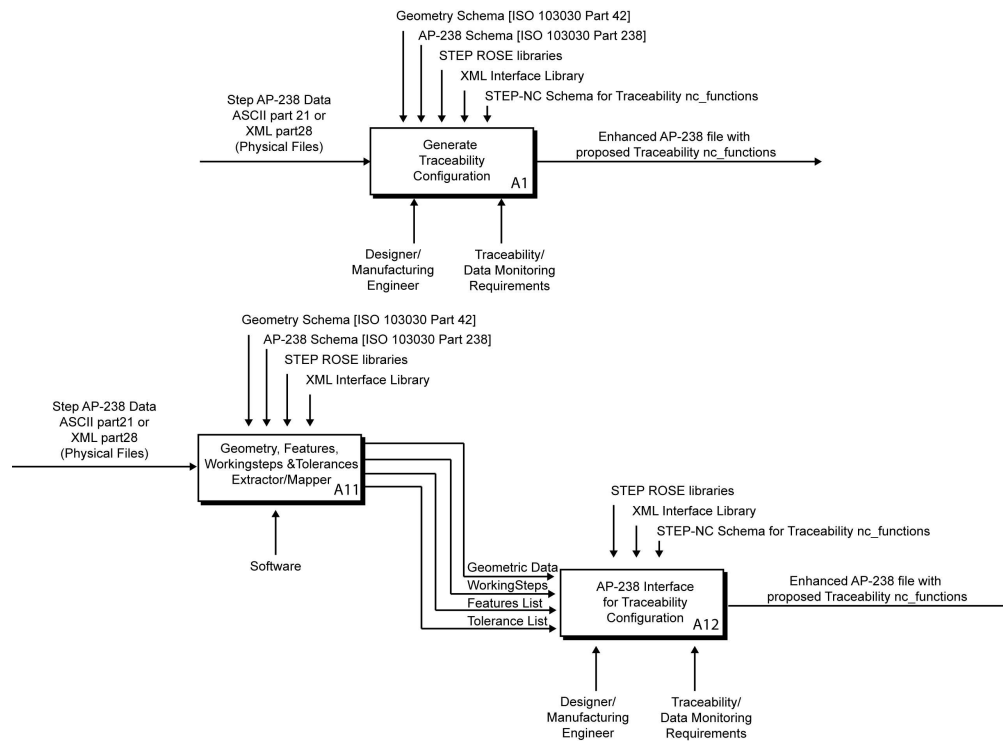


Figure 8. IDEF0 activity diagrams for the implemented traceability and monitoring configuration tool.

217x159mm (300 x 300 DPI)

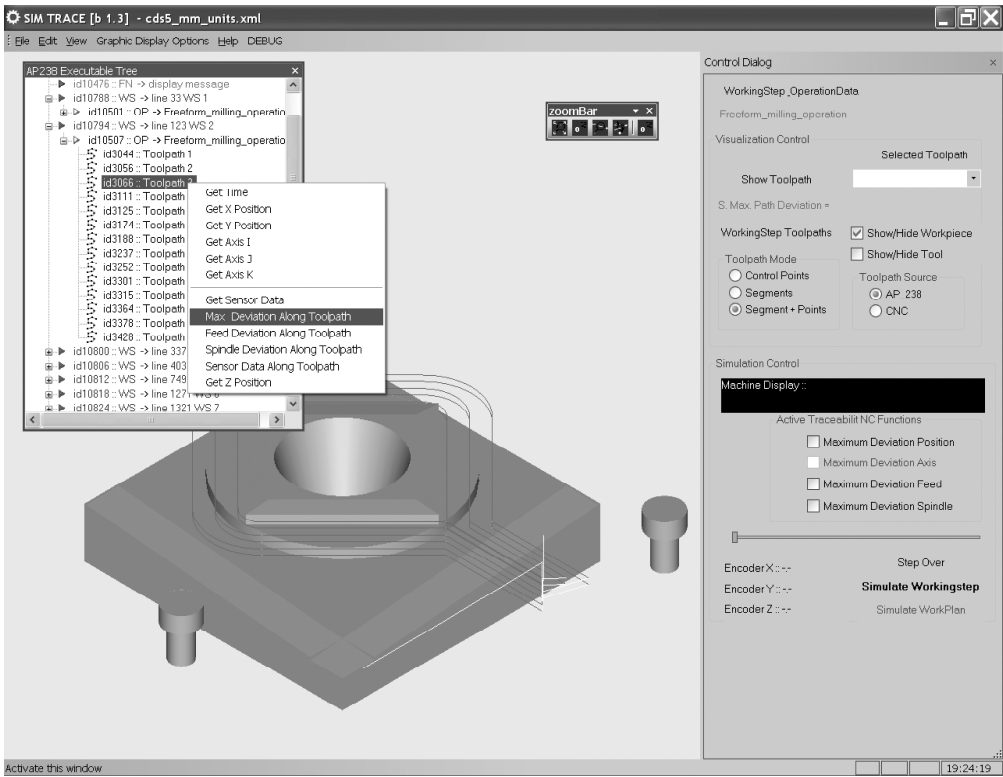


Figure 9. Configuring traceability data requirements over AP-238 data.
451x348mm (300 x 300 DPI)

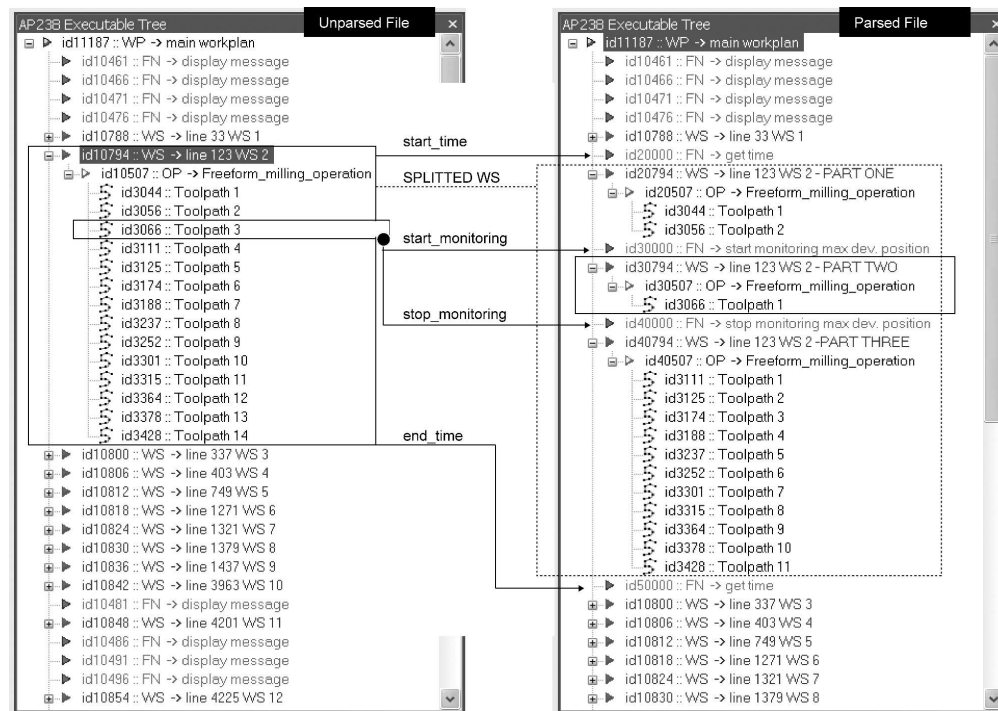


Figure 10. AP-238 executable parsing process.
339x239mm (300 x 300 DPI)

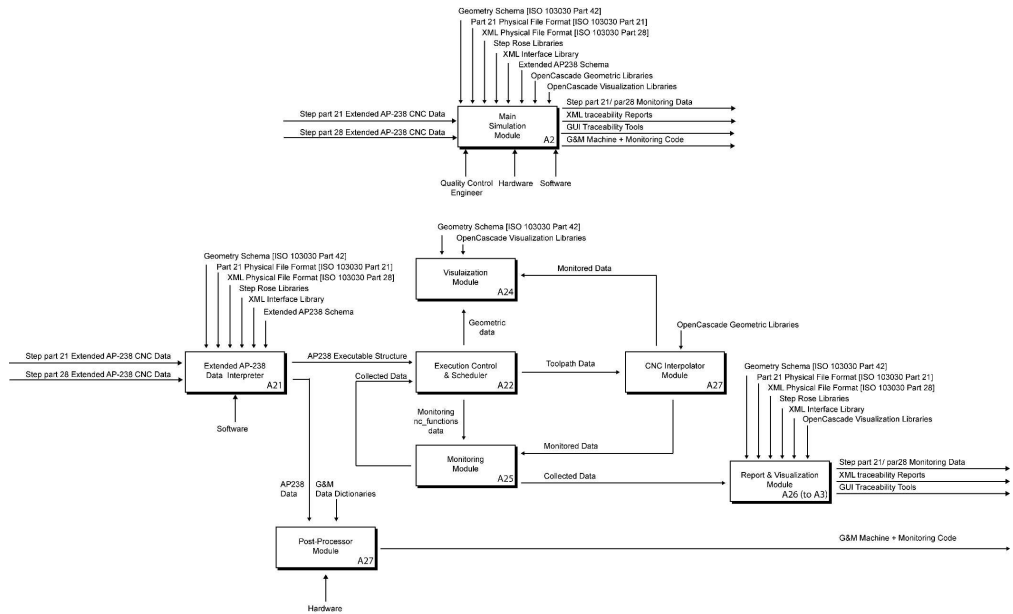


Figure 11. IDEF0 activity diagrams for the implemented traceability and monitoring simulation tool.
356x216mm (300 x 300 DPI)

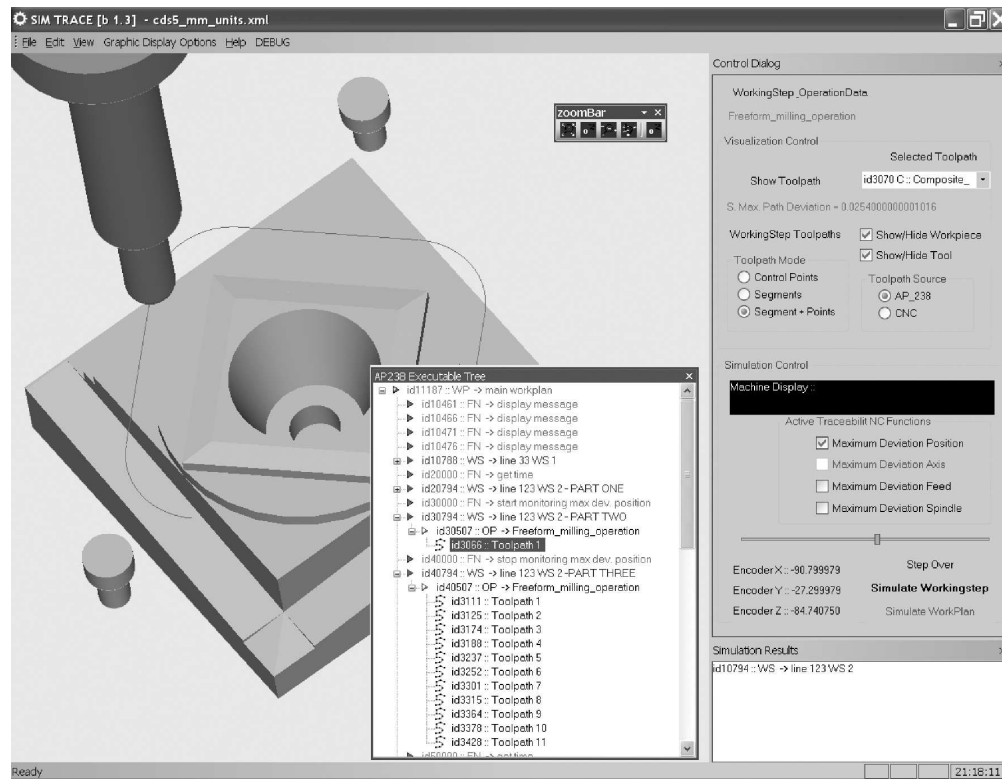


Figure 12. AP-238 CNC simulation.
451x349mm (300 x 300 DPI)

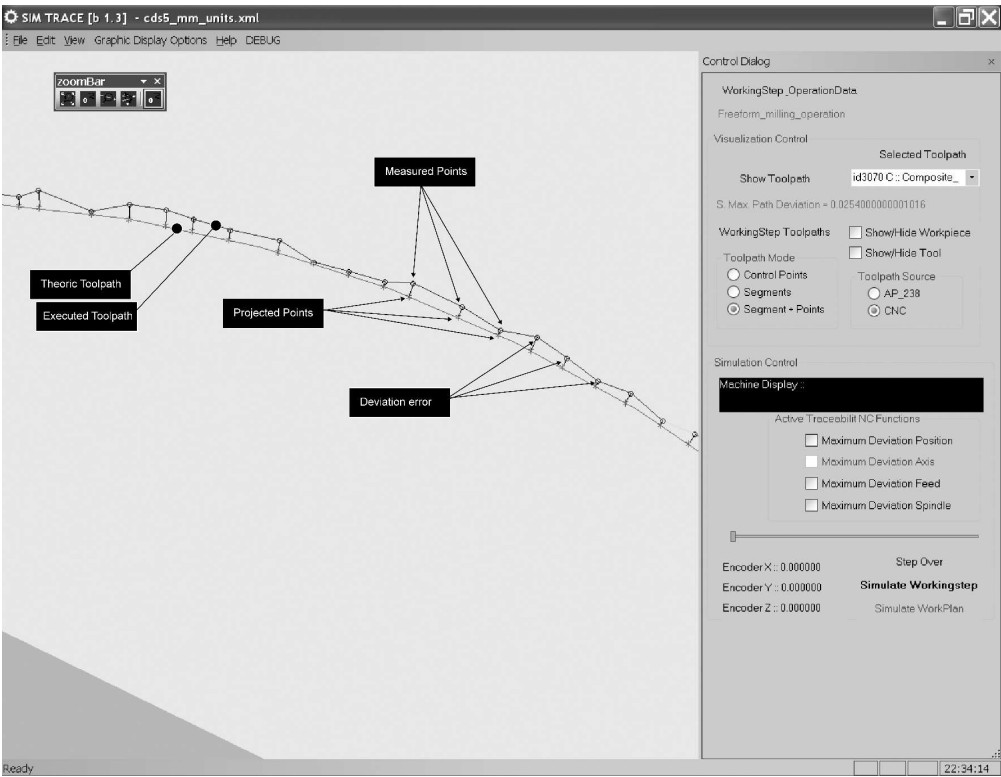
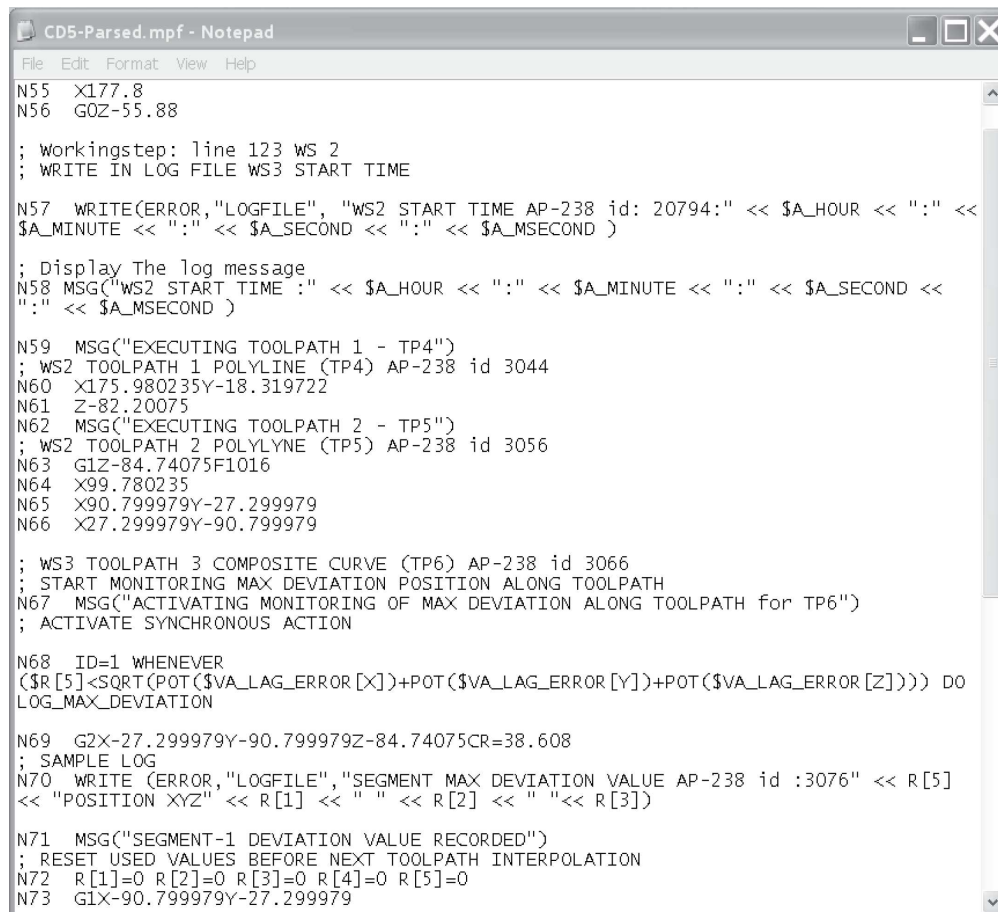


Figure 13. Maximum tool path deviation evaluation.
451x348mm (300 x 300 DPI)



```

CD5-Parsed.mpf - Notepad
File Edit Format View Help

N55 X177.8
N56 G0Z-55.88

; Workingstep: line 123 WS 2
; WRITE IN LOG FILE WS3 START TIME

N57 WRITE(ERROR,"LOGFILE", "WS2 START TIME AP-238 id: 20794:" << $A_HOUR << ":" <<
$A_MINUTE << ":" << $A_SECOND << ":" << $A_MSECOND )

; Display The log message
N58 MSG("WS2 START TIME :" << $A_HOUR << ":" << $A_MINUTE << ":" << $A_SECOND <<
":" << $A_MSECOND )

N59 MSG("EXECUTING TOOLPATH 1 - TP4")
; WS2 TOOLPATH 1 POLYLINE (TP4) AP-238 id 3044
N60 X175.980235Y-18.319722
N61 Z-82.20075
N62 MSG("EXECUTING TOOLPATH 2 - TP5")
; WS2 TOOLPATH 2 POLYLYNE (TP5) AP-238 id 3056
N63 G1Z-84.74075F1016
N64 X99.780235
N65 X90.799979Y-27.299979
N66 X27.299979Y-90.799979

; WS3 TOOLPATH 3 COMPOSITE CURVE (TP6) AP-238 id 3066
; START MONITORING MAX DEVIATION POSITION ALONG TOOLPATH
N67 MSG("ACTIVATING MONITORING OF MAX DEVIATION ALONG TOOLPATH for TP6")
; ACTIVATE SYNCHRONOUS ACTION

N68 ID=1 WHENEVER
($R[5]<SQRT(POT($VA_LAG_ERROR[X])+POT($VA_LAG_ERROR[Y])+POT($VA_LAG_ERROR[Z]))) DO
LOG_MAX_DEVIATION

N69 G2X-27.299979Y-90.799979Z-84.74075CR=38.608
; SAMPLE LOG
N70 WRITE (ERROR,"LOGFILE","SEGMENT MAX DEVIATION VALUE AP-238 id :3076" << R[5]
<< "POSITION XYZ" << R[1] << " " << R[2] << " " << R[3])

N71 MSG("SEGMENT-1 DEVIATION VALUE RECORDED")
; RESET USED VALUES BEFORE NEXT TOOLPATH INTERPOLATION
N72 R[1]=0 R[2]=0 R[3]=0 R[4]=0 R[5]=0
N73 G1X-90.799979Y-27.299979

```

Figure 14. G&M monitoring post-processed sample file.
183x166mm (300 x 300 DPI)

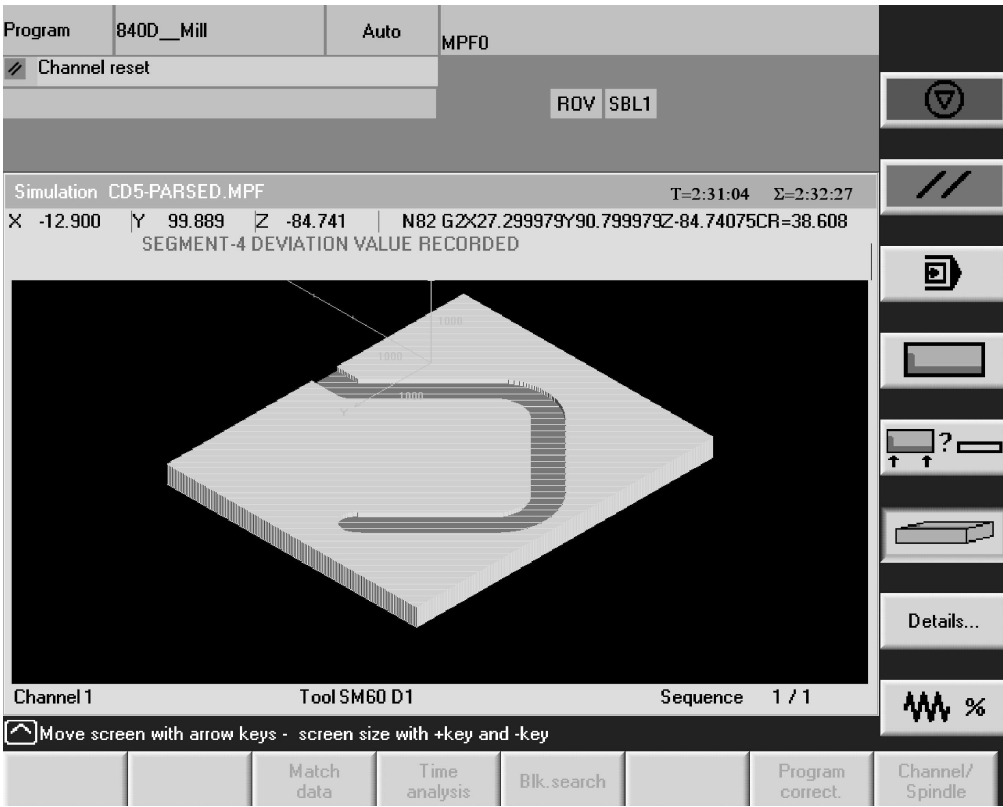


Figure 15. G&M CNC simulation.
 232x186mm (300 x 300 DPI)

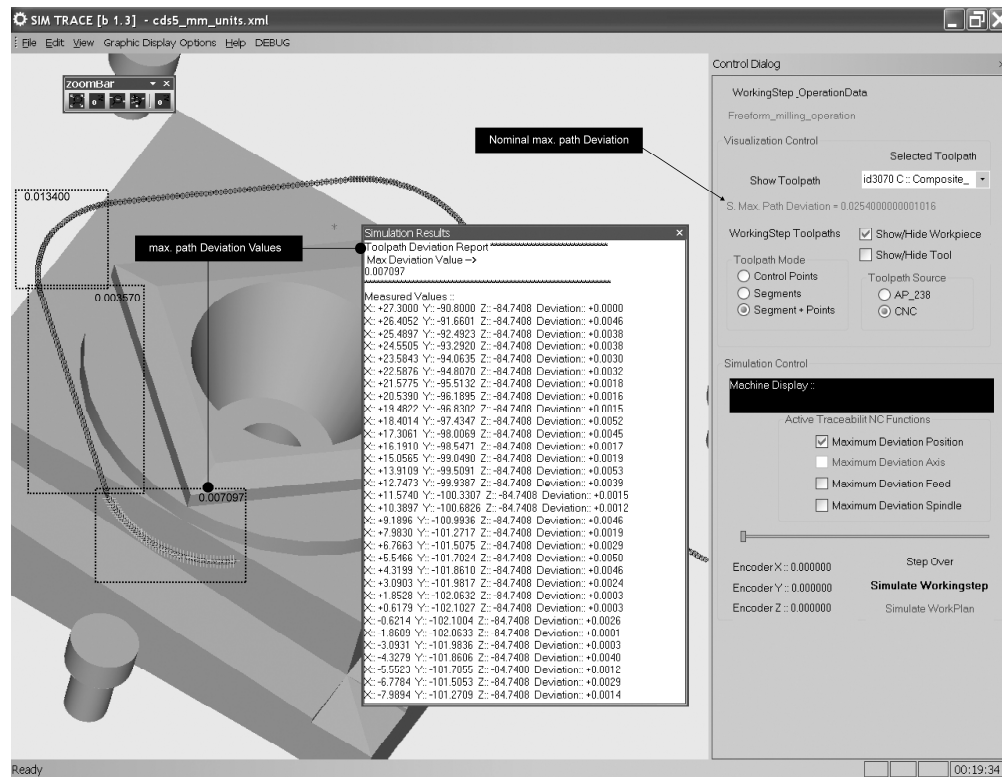


Figure 16. Traceability report.
451x348mm (300 x 300 DPI)