1. Introduction

Traceability is defined as “the ability to trace the history, application or location of an entity by means of recorded identifications.” This definition, incorporated in 1987 into ISO 8402 [1], continues to feature in ISO 9001:2000 quality procedures [2,3]. In practice, traceability of manufacturing processes means recording product manufacturing information regarding raw materials, employees, machines, tools, storage, conditions, and so on [4]. The objective is to be able to react to defects or incorrect behaviour that originated in the manufacturing process for final products [5]. If a part is reported as defective, traceability records can be reviewed to obtain information on, for example, the raw material lot used to make a specific part (trace-back or tracing). Such information may be used to avoid massive product recalls by precisely delimiting individual products made in the same conditions [6]. Globalization requires that traceability activities for data recording and management move from a local view (internal traceability) towards supply chain visibility (external traceability) [7,8]. In spread and global manufacturing environments, main companies (contractor or assembly companies) design the products and are responsible for all their parts, even if manufactured in supplier companies. Redesigning parts, analysing product failures, etc., sometimes needs to be done using information about how they were manufactured on the supplier shop floor. In such scenarios, monitoring and traceability activities such as configuration, shop-floor data collection and data management, are usually performed by different and changing partners. To integrate these activities and enable collaboration across the supply chain, it is necessary to reach an agreement in regard to data format and data exchange protocols [9]. Such integration faces some challenges. The first challenge relates to interoperability issues in complex manufacturing processes with flexible and dynamic supply chains. Plant-level production applications and business systems need to share and exchange information. The second challenge is in regard to data availability overcoming the temporary character of the relationships between enterprises. Traceability data may be required after long periods of time, when some companies may no longer be part of the extended supply chain. A third challenge is the need for common traceability information models that overcome the use of customized company models. Traceability data may be difficult to merge and understand if companies use different data formats and organize information differently. Product data management (PDM) models and manufacturing execution systems (MES) allow traceability tasks to be integrated with other collaborative manufacturing processes [10]. New standards for MES-ERP (enterprise resource planning) integration, such as ISA-95, define information interchange between business functions and manufacturing operation functions, with production process performance data models for feedback from the
MES to ERP systems [11].ISA-95 focuses on MES-ERP communication, but does not cover MES control systems (PLC, CNC controllers, etc.) communication, which rely on standards such as OPC. CNC machine tools are the main components of any manufacturing shop floor [12], and new standards such as MTConnect [13] try to enhance open access to CNC controllers with a neutral definition of information content and access paths. However, all these systems are data access and communication mechanisms that support interoperability. Ultimately, however, the management of multivariable raw data requires metadata to describe the context of the measurements.

Data also has to be identified, indexed and used in subsequent machining process analysis phases. However, the use of G&M codes (ISO 6983) [14]—which is the standard programming language for shop-floor CNC machines—has traditionally limited traceability systems to customized or locally made solutions. Data recorded by these systems—even though they may be related to CNC machining process specifications (a G&M program)—are difficult to relate to other product data because of the lack of such a relationship in G&M-based CNC programs (Fig. 1, top). A new ISO standard, commonly referred to as STEP-NC, is currently being developed to provide a data model for a new breed of intelligent CNC controllers [15]. STEP-NC makes available higher information content to the CNC, describing not only how to make the piece but also what to make [16,17] by incorporating feature-based product knowledge (holes, pockets, etc.) directly in the CNC controller programs as part of the information requirements for NC programming (Fig. 1, bottom).

Although there is a growing interest in incorporating traceability requirements in STEP data models [18], STEP-NC does not define any monitoring or traceability capability. 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 [19,20], and proposals for implementing data acquisition systems with embedded STEP-NC information [21,22]. The problem of the set-up of the recoding process in the CAM to be performed while machining in the shop floor has been addressed at [23] by defining first version of the nc-functions to be added to the new CNC programming standard (ISO STEP-NC). These nc-functions make it possible not only to integrate traceability data with STEP-based PDM data models but also to automate traceability tasks by translating traceability information requirements into shop-floor data access actions [23].

This paper presents a new version of the manufacturing data monitoring functions (nc_functions) and discusses how they may support and supplement other related shop floor data access standards such as ISA-95 and MTConnect.

The rest of the paper is organized as follows. Section 2 describes first new standards—ISA-95 and MTConnect—and Section 2.2 describes the STEP-manufacturing environment and the general structure of STEP-NC programs. Section 3 describes a proposed extension to STEP-NC with new nc_functions for programming process monitoring and traceability. Section 4 presents a prototype implementation scenario based on this extended model and discusses how the standards described in Section 2.1 (ISA-95 and MTConnect) may support and complement STEP-NC in establishing a collaborative manufacturing scenario with programmed traceability and monitoring services. The paper finishes with conclusions and comments on future work in Section 5.

2. New standards for CNC process data

2.1. Standards for enterprise vertical integration

Vertical integration has been tackled in different industrial sectors by defining standard or sector-normalized models [24,25].

Fig. 1. A comparison of ISO 6983 and ISO STEP-NC manufacturing scenarios.
Recently, the ISA-95 standard has attracted a great deal of interest in the automation world by defining a general model for manufacturing operations [26]. The model defines levels that can be mapped to systems (ERP, PDM, manufacturing resource planning (MRP) or MES). A typical physical view of ISA-95 systems is shown Fig. 2. It represents processes as a functional hierarchy model with five levels (0–4), but distinguishes two domains within a manufacturing company: the enterprise domain (Level 4 for business planning and logistics) and the manufacturing domain (Levels 3 and lower for manufacturing operations and control, and process control, whether batch, continuous or discrete). The ISA-95 enterprise/control system standard defines data models as interfaces between Level 4 business functions and Level 3 manufacturing operation functions (Fig. 2) [11,27]. Among information object models that describe production process resources and planning information (“Product Definition”, “Materials”, “Equipment”, “Personnel”, “Production Schedule”, etc.), ISA-95 also defines a “Production Performance” model for Level 3 (manufacturing) information to be provided as feedback to Level 4 (business). Some software vendors such as Fanuc, Rockwell Automation and Siemens have implemented a common component framework across their respective products to create a common data model based on the terminology, object models and XML schemas described in ISA-95 specifications [11]. However, ISA-95 focuses on the exchange between MES (ISA-95 Level 3) and ERP (ISA-95 Level 4), but communications between Level 3 and controllers rely on custom/brand mechanisms [20] or on other standard protocols, such as OPC [28]—oriented to data access and transfer but not to information content description (Fig. 2).

New open communication standards such as MTConnect allow devices, equipment, and systems to output data in an understandable format that can be read by any other device [13]. MTConnect is an open and extensible protocol that allows the exchange of dynamic sensor data, configuration data, and control information among MTConnect-compliant machines, software applications and controllers. MTConnect messages are encoded using XML, with a self-describing structure for each data item, as illustrated in the example in Fig. 3.

All these systems are data access and communication mechanisms that support interoperability. Ultimately, however, the management of multivariable raw data requires metadata to describe the context of the measurements. There is little guidance in ISA-95 on how this data can be identified, indexed, searched or otherwise used, but means should be provided to generate and maintain metadata associated with measurements [11]. The specification for the machining process is the metadata needed to interpret data coming from the process. Since the new standard STEP-NC completely documents the manufacturing process, it may be used as the metadata to describe the same process.

2.2. Standards for CAD/CAM/CNC integration: STEP v STEP-NC

STEP (ISO 10303: STandard for the Exchange of Product data models) is widely used as a technology for the file-transfer of design data [29]. For example AP-203 (application protocol) is used to exchange design data among CAD systems and between CAD and CAM systems [30]. New advances in CAD/CAM/CNC collaborative systems are promoting the development of new models (application protocols, AP’s) to support PDM (product data management) technologies, real-time communication and collaboration during different phases of product development.

STEP 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 STEP manufacturing suite of application protocols (SMS AP’s) [31].

Fig. 4 shows a manufacturing scenario for mechanical parts and assemblies based on design features using the SMS AP’s. 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 Fig. 4 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 [30]. 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 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 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 feature machining information.

STEP-NC is an object oriented language with a richer interface than ISO 6983 for transferring information between CAD/CAM programming systems and computerised numerical controllers [35]. Although it is still globally used, ISO 6983 has become a bottleneck because the information passed to the CNC controller is limited to how-to-make information (Fig. 1, left), excluding information on the actual workpiece, and also because of data model non-compliance through CAD, computer-aided process planning (CAPP), CAM and CNC [36]. STEP-NC 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. Two versions of STEP-NC are being developed. The first is the ISO 14649 [37] to define the high level information requirements for CAM and CNC data interchange. The second STEP-NC version, the ISO 10303 AP-238, has adopted ISO 14649 models to define the implementation oriented models to build a common language with other STEP CAD, CAM, CAPP systems.

The STEP-NC information requirements cover geometric information (workpiece, geometric tolerances, tool path, manufacturing features...), control structures for the program execution (workplan), technological information (process data for milling, tools for milling...), and data management (Fig. 5, middle top). A workplan specifies the information requirements to
Fig. 5. Step-NC structure.

Fig. 6. Traceability nc-functions data structures.

Table 1
Traceability nc-functions.

<table>
<thead>
<tr>
<th>Group</th>
<th>Function name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td><code>get_time get_operator get_machine get_tool</code></td>
<td>It gets the present time in the controller. It gets the identification of the machine operator. It gets the identification of the machine being used. Of the coolant in use. It gets the identification of the tool in use. It gets the identification number of the raw workpiece. It gets the actual value of a sensor</td>
</tr>
<tr>
<td></td>
<td><code>get_coolant get_raw_material get_sensor_data</code></td>
<td></td>
</tr>
<tr>
<td>II/III</td>
<td><code>start_measuring_maximum_deviation_position_along_toolpath</code></td>
<td>It starts the measurement of the maximum deviation of the tool position along the tool path. A unique deviation value, the maximum, is saved for each segment. It stops the process. Parameter function. It starts the monitoring of a sensor value identified by the “sensor_id” attribute value. It stops the monitoring of the sensor identified by the “sensor_id” attribute value</td>
</tr>
<tr>
<td></td>
<td><code>stop_measuring_maximum_deviation_position_along_toolpath</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>start_monitoring_sensor_data_along_toolpath</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td><code>stop_monitoring_sensor_data_along_toolpath</code></td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td><code>start_get_operator start_get_tool start_get_override</code></td>
<td>They start concurrent—with machining—register of asynchronous events: change of operator, un-expected change of tool, manual tool speed overrides, controller events (alarms, stops, etc.). It starts the register of starting and finishing time for each program action: workingstep, nc_function, . . .</td>
</tr>
<tr>
<td></td>
<td><code>start_get_controller_events . . . start_get_time</code></td>
<td></td>
</tr>
<tr>
<td>V</td>
<td><code>set_alarm input</code></td>
<td>It specifies an alarm number. It is similar as “DisplayMessage” but it also asks for an input</td>
</tr>
</tbody>
</table>
describe the control flow of the machining program, as well as some non-machining actions that may be performed by NCS: workingsteps, nc_functions and program structures. 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. Nc_functions describe switching operations or other non-interpolating machine functionality such as tuning the coolant on. Fig. 5(top) shows an EXPRES-G diagram of the STEP-NC information model general structure, where executable unit of functionality hangs from the workplan entity. The figure also shows (middle shaded box) a STEP executable example in a pseudo STEP format (STEP part 21, the STEP format for persistent data). In the example, a STEP-NC CAM file is a project with feature geometric information and a workplan to perform them. The workplan has an nc_function call to load the appropriate tool and a workingstep with a sequence of pre-calculated toolpath segments.

3. Data model for traceability activities

ISA-95 covers vertical integration, which refers to integration from the office through to the factory floor (integration of the
enterprise domain with the control domain), typically within a single company or site. This is the opposite of supply chain integration (horizontal integration), in which different companies or sites mutually respond and collaborate as each other's clients or suppliers—scenarios covered by standards such as STEP. A STEP-NC program is a neutral CNC process representation for a horizontal integration scenario. It is the standard format for representing information automatically as actions in a CNC system, but it does not specify how that system has to work; in other words, it is a passive model and does not include any intelligent functionality [38]. STEP-NC model extensions are being developed to support neutral exchange of requirements for new functionalities such as dynamic compensation [39], machine capabilities specification [40], etc. This section describes new STEP-NC functionalities for programmable traceability and programmed process monitoring, while Section 4.4 will discuss how this proposal may work with other standards for vertically integrated data access, such as ISA-95 and MtConnect.

Table 1 describes some of the proposed traceability nc functions. These new traceability nc functions can be categorized in five groups. Fig. 6 illustrates the EXPRESS-G model of some the nc functions already in the standard (display_message, load_tool), and also some of the new proposed nc functions for each group.

Group I functions are used to enrich the range of actions that can be taken, as for instance set_alarm function and input function—similar to display message but force the operator to enter some data.

Group II functions define traceability actions that must be performed by the controller concurrently with other CNC machining operations. They 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 Fig. 6. 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 parameterization 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. The functions and the data structure are designed to save a limited amount of data. In other words, functions do not retain all the monitored values in memory, only the maximum monitored value for each segment. However, more detailed data, as well as the data in memory, may be stored following another format, as our next implementation example will illustrate.

Group III functions are the same as group II but with the extended ability to perform actions if a threshold is crossed. 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. If a workplan and a threshold value are specified, then the group II function becomes a group III. Fig. 7 illustrates a STEP-NC example program with group III nc functions (left), and also shows a graph representing the execution steps (middle) and the resulting traceability data (right). Group I, II and III nc functions have been described in detail in [21].

Group IV functions are similar to group I ones but with global program scope as they get a series of values concurrently with the machining process. These functions are to start the recording of "asynchronous events". Asynchronous events are machining process condition changes that are not pre-specified in the machining program, but that they may occur while machining, as for instance operator log-in/log-out, controller events such as mode changes (automatic to stop changes, resume operation event, etc.). Fig. 8 is an example with two of these functions: start_get Override, which stores the time and the program object in execution (workstep, feature) when the operator makes a manual change in the tool speed, and start_get_controller_events which make the same kind of register when the controller has an internal alarm.

Finally, to support group III functions, group V are defined to enrich the range of actions that can be taken, as for instance set_alarm function and input function—similar to display message but to force the operator to enter some data.

![Fig. 10. Traceability data flow and applications.](image-url)
4. Prototype implementation

A prototype STEP-NC manufacturing example scenario has been implemented using the traceability nc_functions proposed in Section 3.

Fig. 9 identifies—in a high-level AAM (application activity model)—the main manufacturing traceability tasks—configuration (activity A1), data collection (activity A2) and data reporting (activity A3).

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 (Fig. 10). Through the downstream data flow, traceability configuration requirements are communicated from qualified data sources to the shop-floor (left part in Fig. 10). These requirements are interpreted on the shop-floor and result in automated data collection (bottom part in Fig. 10). The data can then be communicated back by following an inverse upstream data path to the configuration levels (right part in Fig. 10).

Three basic interfaces have been done—shadow boxes in Fig. 10. STEP Tools NC-Explorer and STEP-NC dll have been used as the core tools to process STEP-NC files and programming the interfaces. These interfaces have been coded using the C++ programming language and STEP Tools Developer 11 libraries [41] under MS Visual Studio .Net. The first interface is used to set up monitoring information requirements in an AP-238 design (interface I in Fig. 10). As result, an extended AP-238 design file is generated including new traceability nc_functions in the executable structure. This interface has been coded using StepTools resources and it has been described in detail in [21]. The second module is a native STEP-NC CNC simulator which it is used to check and simulate the execution and the data access (interface II in Fig. 10). Third interface of computer aided quality CAQ (interface II in Fig. 10) is a native graphic interface to graphically browse and view monitored data coming from the CNC, and it will be described in Section 4.3.

4.1. Requirements configuration

First activity of traceability configuration is the specification of data access requirements—activity A1 in Fig. 9. Configuration means mapping data collection requirements into machine understandable information—a program. This allows to specify which data must be collected during shop-floor machining (semantic data scope), but for process automation, configuration must also indicate when to start and finish collecting the data (temporal data scope). Both objectives can be covered for STEP CNC machining environments with the proposed nc_function approach.

The configuration interface 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. Fig. 11 shows how the configuration application. In the figure, the workpiece is described as a freeform feature to be machined, so the sequence of workingstep 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 #8157 in Fig. 11) are configured as requirements to be monitored.

After selecting the traceability requirements, the module is ready to translate the selected user requirements into AP-238 compliant traceability nc_functions. As result, extended code is generated for the corresponding nc_functions and inserted in the
AP-238 files to generate “extended” versions of the AP-238 files with process monitoring capabilities—the output of activity A1 in Fig. 9 and the output file of Interface I in Fig. 10.

4.2. Data access

Extended files are sent to the shop floor to make the part (activity A2 in Fig. 9). In the prototype implementation, the machining process is simulated by a native STEP-NC simulator module (Fig. 10, interface 2). The simulator examines and checks the AP-238 executable structure to perform a step by step program execution as a native hardware STEP-NC controller would do it. If a feature has been selected to be monitored, the corresponding nc_functions are inserted in the executable structure and goes into monitoring mode.

For instance, if execution time has been programmed (at CAM level) to be monitored, when the simulator finds the “get_time” extension code, it get the current time and updates the value of the nc_variable associated to this nc_function. For the case of monitoring variables along a tool path, when the controller finds a Start_measuring_.nc_function, it boots a concurrent process to access data in predefined points of each segment of the tool path. These data may be used at machining time (run time logged data in Fig. 10), or more detailed data may be stored in a file (persistent logged data in Fig. 10) following any other format (as Section 4.4 suggests) to be used off-line.

4.3. Off-line data analysis

These data are communicated back through the supply chain to be visualized, processed, audited and stored (Fig. 10, bottom data flow) for quality control purposes or to support other advanced services, as for instance machining process optimization. The implicit linkage between collected traceability data and CAM product data makes it possible to understand the collected traceability data by identifying the manufacturing process that they document (AP238 metadata and XML ISA95 process performance data in Fig. 10). The value of traceability information is dependant on its relevance, completeness, accuracy, availability, veracity and accessibility. Efficient traceability implementations should provide exactly the right information at the right time, and auditing is an essential tool for the outsourcing company to examine traceability data and compare it with the traceability requirements (audit step in Fig. 10), thereby detecting data that has not been completed or correctly recorded by the manufacturing facilities, and checking the value of traceability information.

Fig. 12 shows an example of data visualization for the maximum deviation position along the tool segments of a workingstep. The figure shows that it is possible to select a specific workingstep (id8157 in the figure), in the CNC program executable structure and check the corresponding recorded value. The toopath of the workingstep is represented in the scream, and it is possible to graphically select each one of the segments to visualize the recorded value of the feed rate deviation during machining.

4.4. Discussion

With traceability and monitoring nc_functions inserted in a CAM part program, traceability and monitoring actions can be programmed and automatically executed by the controller. If data records are linked to a standard machining process specification that is understandable worldwide (the STEP-NC part program), records may be understood and correctly analysed. However, the
The traceability nc_functions proposal has two main concerns. One is the standard and neutral way of referencing controller variables or sensors, in view of the fact that any controller may have their own particular variables and new ones may appear with the development of new machines. Although generic nc_functions have been defined for accessing any kind of variable—for example, start_monitoring_sensor_data_along_toolpath (as in Fig. 6)—by specifying the sensor or variable name with a string parameter (Sensor_Id in Fig. 6), it is still necessary to define a set of valid names. For this purpose, new standards such as MTConnect could be used, as they define data access paths and names for controller variables and sensors (see Fig. 3).

A second problem is the traceability data logging strategy: whether free format (option A in Fig. 13) or standard format, but if standard, whether with STEP format (option B in Fig. 13) or following other standard format (option C in Fig. 13).

Choosing option B would imply developing new STEP data model for traceability data. However, there are already standards that may be used for this purpose. ISA-95 has defined models to store data recorded during the manufacturing process to be used in production process control and in optimization and quality control activities such as traceability—namely, the production performance model (XSD schemas for XML data files) [42,43]. These can be used to store data coming from the execution of the traceability nc_functions, provided that the STEP-NC part program is also referenced and provided that each data record is linked with the corresponding STEP-NC object (a nc_function id, a tool path id, a feature id, etc.). Fig. 14 shows an example of traced data following an ISA-95 schema linked with a STEP-NC process description.
example of an ISA-95 production performance structure used to store data coming from the execution of a STEP-NC file (AP-238), maintaining a link between them and relating the data records in the XML file to the corresponding objects in the AP-238 file.

5. Conclusion

The research presented in this paper has shown that traceability activities (configuration requirements, set-up, data access, data storage, data analysis) can be integrated in STEP-NC collaborative CAD/CAM/CNC chains. In our proposed approach, CNC controllers can interpret and automatically execute programmed monitoring and traceability commands (traceability nc_function), allowing automation of main traceability activities performed in a collaborative manufacturing scenario, as follows:

- Automatic traceability requirements communication to the shop-floor and automatic interpretation by the CNC controller (set-up). It is not always necessary or even desirable to collect machining data about all features of the part since most critical manufacturing errors can be found by checking a selected group of features. The nc_functions approach is a way of programming traceability and monitoring actions for a selected group of features or toolpaths.
- Automatic traceability data collection on the shop-floor. Automatic data access ensures data reliability. If data is reviewed in the future, there will be no concerns about the method for retrieving that data in the past (probably from a subcontractor shop-floor).
- Automatic communication, review and management of the traceability data collected on the shop-floor. STEP-NC documents the manufacturing process and may be used as metadata for other standards to attribute significance to their data.

Traceability and monitoring solutions discussed in the paper (new nc_functions) are being examined by the ISO TC184/SC4 WG3 T24 STEP-manufacturing committee for possible inclusion in future editions of ISO 10303 STEP AP-238. Currently, preliminary versions of group II and group III traceability nc_functions monitor manufacturing actions with an explicit toolpath. However, if there is no explicit toolpath it is up the controller to decide its own toolpath based on the feature information of the STEP-NC file. Future work on tackling this kind of situation will define extensions for tracing semantic operations (for example, mill a pocket). Another STEP-NC extension is in development for metrology and in-process measurement is an area for further work. The group II functions approach (or similar ones) can be used to make in-cycle measurements while the group II functions approach can be used to make in-process measurements. The group III functions approach can be used to trigger corrective actions. How to use group III to correct errors is yet another area for future research.

Finally, the paper has shown how standards like ISA-95 and MTCOnnect may support, rather that interfere with, STEP-NC in addressing the two main problems of the traceability scenario: how to specify valid machine variables or sensor names and how to define standard data formats for persistent storage and communication of traceability records. The former may be supported following the MTCOnnect approach for accessing machine variables, and the latter may be resolved by using the ISA-95 standard performance model but incorporating links between ISA-95 data records and the corresponding STEP-NC file entities. From the ISA-95 point of view, metadata is necessary to interpret and analyse data coming from the machining shop-floor, and a standard STEP-NC file may play this role.

Acknowledgements

The author wishes to acknowledge Martin Hardwick at Rensselaer Polytechnic Institute for his assistance and research into STEP.

References


Dr. Julio Garrido Campos is an associate professor in the department of Automation and Systems Engineering of Vigo University (Spain). He got his engineering degree and PhD degree (1999) at Vigo University, after being for one year (1995) a research assistant at Computer Science Department of Rensselaer Polytechnic Institute (Troy, NY). His current research interest includes industrial traceability for collaborative environments, traceability automation, ISO 10303 STEP standard, etc. Also, as a member of the ISO TC184/SC4 WC3 T24 STEP manufacturing group, he has leaded the efforts of this ISO committee to incorporate traceability into the ISO 10303 AP-238. He is the author of papers and articles on engineering automation and STEP, and he has also worked as a consultant in many industrial projects regarding traceability and industrial automation in the automotive sector.

Luis Rodriguez Miguez is a PhD student at department of Automation and Systems Engineering of Vigo University (Spain) where he has got his Telecommunication Engineering degree. His research interest is in models for collaborative manufacturing, STEP manufacturing suite, etc.
免费论文查重：http://www.paperyy.com
3亿免费文献下载：http://www.ixueshu.com
超值论文自动降重：http://www.paperyy.com/reduce_repetition
PPT免费模版下载：http://ppt.ixueshu.com