Published November 1, 2002. Distribution restricted to Sponsors until February 1, 2003.

WHITE PAPER

Auto-ID's Three R's: Rules and Recipes for Product Requirements

Duncan McFarlane, James Carr, Mark Harrison, Andrew McDonald

AUTO-ID CENTRE INSTITUTE FOR MANUFACTURING, UNIVERSITY OF CAMBRIDGE, MILL LANE, CAMBRIDGE, CB2 1RX, UNITED KINGDOM

ABSTRACT

This paper introduces the related concepts of rules and recipes as a means of capturing product requirements in an Auto-ID systems environment. A rule represents an action taken on the basis of a logical condition, and a recipe a structured sequence of rules for the completion of an operation. In this paper we specifically associate the idea of a recipe with products and contrast it to the "machine instruction" representing the sequence of steps carried out on a specific machine or resource. The uses of recipe information in the supply chain – and in fact the entire product life cycle – are numerous, ranging from supporting out-sourced production, to distribution instructions for perishable products, to directing maintenance procedures. In the context of an Auto-ID system, rules and recipes form part of the information set that a tagged product may be connected to, in addition to basic parameters and history data. Clearly such information will be closely linked to the Physical Mark-up Language developments currently underway.

The aims of this introductory paper are to:

1. To clearly define rules, recipes and machining instructions
2. To establish their industrial relevance and in particular the relevance within the Auto-ID project.
3. To provide an initial indication as to how rule and recipe information will be integrated within an Auto-ID environment

Additionally a simple classification of recipes is provided and a summary of related standards included.

Published November 1, 2002. Distribution restricted to Sponsors until February 1, 2003.

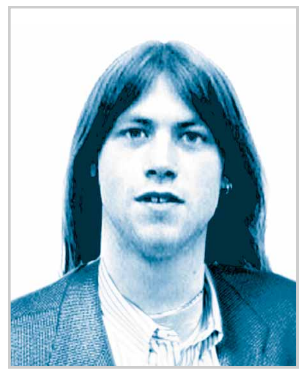
WHITE PAPER

Auto-ID's Three R's: Rules and Recipes for Product Requirements

Biography

Duncan McFarlane
Research Director Europe

Duncan McFarlane is a Senior Lecturer in Manufacturing Engineering in the Cambridge University Engineering Department. He has been involved in the design and operation of manufacturing and control systems for over fifteen years. He completed a Bachelor of Engineering degree at Melbourne University in 1984, a PhD in the control system design at Cambridge in 1988, and worked industrially with BHP Australia in engineering and research positions between 1980 and 1994. Dr McFarlane joined the Department of Engineering at Cambridge in 1995 where his work is focused in the areas of response and agility strategies for manufacturing businesses, distributed (holonic) factory automation and control, and integration of manufacturing information systems. He is particularly interested in the interface between production automation systems and manufacturing business processes.



James Carr
M. Eng. Candidate

James Carr recently graduated from Cambridge University with B.A. and M.Eng. degrees in Manufacturing Engineering. As part of his final year, he completed a thesis on the application of recipe standards within Auto-ID based information environments.

Published November 1, 2002. Distribution restricted to Sponsors until February 1, 2003.

WHITE PAPER

Auto-ID's Three R's: Rules and Recipes for Product Requirements

Biography

Mark Harrison
Research Associate

Mark Harrison is currently collaborating with the Auto-ID Centre lab in Cambridge on the development of a PML server and web-based graphical control interfaces. In 1995, after completing his PhD research at the Cavendish Laboratory, University of Cambridge on the spectroscopy of semiconducting polymers, Mark continued to study these materials further while a Research Fellow at St. John's College, Cambridge and during 18 months at the Philipps University, Marburg, Germany. In April 1999, he returned to Cambridge, where he has worked for three years as a software engineer for Cambridge Advanced Electronics/Internet-Extra, developing internet applications for collaborative working, infrastructure for a data synchronisation service and various automated web navigation/capture tools. He has also developed intranet applications for his former research group in the Physics department and for an EU R&D network on flat panel displays.

Andrew McDonald
Manager, Unilever North America

Andrew holds the position of Global Automation and Control Technology Manager, Unilever Home and Personal Care North America and is located in Trumbull, Connecticut, USA. He graduated from Aston University, UK in 1986 BSc Electrical and Electronic Engineering, and is a Chartered Engineer and a Member of the Institution of Electrical Engineers. His early career was spent with electrical and systems integrators before undertaking control engineering roles on various waste management and decommissioning projects within the UK Nuclear Industry. He led flywheel energy storage system product development programme for an international energy systems company, focussing later on batch process control in the pharmaceutical and consumer packaged goods industries. Andrew is currently responsible for the management of Unilever's strategic programme relating to process control and automation.

Published November 1, 2002. Distribution restricted to Sponsors until February 1, 2003.

WHITE PAPER

Auto-ID's Three R's: Rules and Recipes for Product Requirements

Contents

1. What are Rules and Recipes? 4
   1.1. Definitions and Clarifications 4
   1.2. Product Recipes Vs Machine Instructions 4
   1.3. Outline of the Report 5
2. The Role of Rules and Recipes in the Product Life Cycle 5
   2.1. Supply Chain Rules and Recipes Information Requirements 6
   2.2. Impact of Recipe Information 6
3. Relevance of Rules and Recipes for Auto-ID Systems 7
   3.1. Introduction 7
   3.2. Recipes and a Single Identification Standard 7
   3.3. Recipes and Product Data 9
   3.4. Recipes and "Intelligent Products" 9
4. How will Recipes be Used in Auto-ID Environments? 10
   4.1. Introduction 10
   4.2. Recipes and PML 10
   4.3. Summary 12
5. Key Issues 12
   5.1. Recipe Classification 12
   5.2. Industrial Recipe Standards 13
6. Conclusions and Next Steps 15
   6.1. Conclusions 15
   6.2. Next Steps 16
7. References 17
Appendix 1 18

1. WHAT ARE RULES AND RECIPES?

This paper establishes the role of **rules** and **recipes** within an Auto-ID environment and establishes the way in which they address product requirements in a clear and consistent manner. We will demonstrate that it in the same way that it is natural for a physical product to be electronically linked (via RFID tags/readers and a network routing system) to information such as key parameters and history data, it is also important for a product to be linked to rule and recipe information which directs its passage through the supply chain.

1.1. Definitions and Clarifications

The definitions proposed here are shamelessly adapted to suit the supply chain context of the Auto-ID Centre.

(Conditional) Rule: an action to be taken based on the outcome of a logical condition.

The conditional outcome and the subsequent action may be dependent on specific parameters.

(Product) Recipe: a structured sequence of data, rules and actions that help to define the requirements for a product to be manufactured, distributed, retailed, used and disposed of.

For the purposes of this report, we categorise the types of information held in a recipe into the following categories:

- **Recipe Administration** – transaction details, ownership, version number
- **Geometrical Representations** – product description data
- **Rules/Constraints** – actions and logical conditions and constraints on operations from product perspective
- **Processing Steps** – key processing steps required to be performed

1.2. Product Recipes Vs Machine Instructions

In this report we clearly differentiate between two classes of instructions relevant to supply chain operations, namely (product) recipes and so called **machine instructions**.

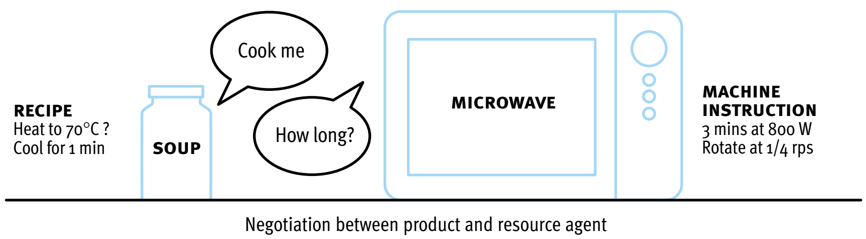
Machine Instruction: a structured sequence of data, rules and actions that is defined to be executed on a specific machine or classes of machines.

The difference here is that the (product) recipe is essentially **machine or resource independent** – it is formed without direct consideration of the specific resources that will carry out the steps of the recipe. In contrast the machine instruction is entirely **machine or resource dependent** having no other use other than to describe the operations on a specific machine. The difference is one of subject/object or transformee/transformer.

Example 1: That Microwave Oven!
A favourite Auto-ID example is the heating of ready meals in microwave ovens. Figure 1 illustrates how the recipe for a jar of fresh soup itself is independent of any particular type of heating appliances. Either through an interpretive panel on the oven or by human interpretation this recipe is converted into a set of steps for the microwave to follow so that it cooks the soup in a way that does not breach any of the constraints described in the recipe. The same recipe could also be used to generate an alternative set of machine instructions for cooking in a conventional oven.

Figure 1

RECIPE VS. MACHINE INSTRUCTIONS (DOMESTIC USE EXAMPLE)



1.3. Outline of the Report

The uses of recipe information in the supply chain – and in fact the entire product life cycle – are numerous, ranging from supporting out-sourced production, to distribution instructions for perishable products, to directing maintenance procedures. In the context of an Auto-ID system, rules and recipes form part of the information set that a tagged product may be connected to, in addition to basic parameters and history data. Clearly such information will be closely linked to the Physical Mark-up Language developments currently underway.

2. THE ROLE OF RULES AND RECIPES IN THE PRODUCT LIFE CYCLE

Associating rules and recipes directly with a product can lead to an alternative way in which supply chain information is managed, with a product or an order "carrying" with it the basic requirements for its creation and useful life.

In this section we overview classes of recipe information associated with each stage of the product life cycle illustrated in Figure 2.

Figure 2

PRODUCT LIFE CYCLE

2.1. Supply Chain Rules and Recipes Information Requirements

A product has associated with it a single recipe with different parts – each linked to the different stages of its life cycle. The table in Table 1 illustrates the type and style of information that is likely to be found in the different parts of a product recipe.

Table 1

TYPE OF INFORMATION WITHIN A RECIPE

PART OF PRODUCT RECIPE	TYPE/STYLE OF INFORMATION
Recipe Administration	Transaction details Recipe ownership Priority coding
Manufacture	Geometrical representations Processing Inputs and Outputs Rules/Constraints and Options Processing Instructions
Distribution	Rules/Constraints
Retail	Rules/Constraints
Usage/Maintenance	Rules/Constraints Processing Instructions
End-of-Life	Rules/Constraints Retirement Inputs and Outputs Processing Instructions

Importantly we note here that there is a set of instruction information associated with a product type – and specifically for an individual item in that product class – that is associated with maintaining the status of different recipe stages and the resulting transactions carried out when the recipes have been fulfilled.

We emphasise that this table represents an initial listing of identified information requirements which will be revised and expanded during the course of this work.

We now consider the possible benefits of being able to directly associate this instruction information directly with the products on which the instruction are executed.

2.2. Impact of Recipe Information

The deployment of recipe information as a means of standardising the operations applied to a product is not new and has been applied in manufacturing within the process and electronic industries (See for example, Standard Recipe File Format – SRFF – in electronics assembly or the Batch Recipe Standard ISA S88 – in the process industry). The differences in what we are considering here are:

– The recipe information is entirely product oriented and is “owned” by the product or order owner.
– Recipes are being considered at each stage of the product life cycle, not simply the production stage.
– Will are seeking environments in which the recipe is interpreted and executed in an automated manner.

Some of the current and future gains from the availability of a product-oriented recipe available as part of the physical product are:

1. **Product Consistency** – the application of a standard recipe for all products of a particular class provides a means of ensuring consistency.
2. **Quality Control** – monitoring and recording of quality data history in direct comparison with a record of processing instructions enables a clear, traceable mechanism for isolation and resolving quality assurance threats.
3. **Distribution of Operations/Outsourcing/3rd Party Involvement** – a key rationale for well defined product recipes is the benefit they provide in terms of enabling simple outsourcing. Equipment independent recipes allow brand owners greater flexibility in terms of selecting manufacturers. It also allows manufacturers greater internal flexibility in terms of selecting a manufacturing site, or even specific cells or product lines within that site. The product supplier can post the order with recipe on an internet portal for bid by qualified sourcing units. The recipes, therefore, allow for a much greater mobility of knowledge and flexibility in terms of available manufacturing environments.
4. **Product Use and Maintenance** – usage and maintenance rules and instructions become directly available during the use and maintenance phases of the product life cycle.
5. **Customisation of Products** – in a highly customised production, delivery or retail environment where circumstances alter the way in which a product is handled, the availability of item-specific instructions directly linked to the physical product leads to a smaller scope for error.
6. **Theft Prevention** – rules for the detection of theft of specific product types can be included in the product recipe portfolio.

3. RELEVANCE OF RULES AND RECIPES FOR AUTO-ID SYSTEMS

3.1. Introduction

The Auto-ID Centre's goals are to produce a standards and infrastructure for the networking of physical objects – with particular reference to the supply chain. To underpin these goals a system's architecture has been proposed in which objects (products) with a unique identification number are connected via RFID communications to one or more databases holding information relevant to that product. The development of a Physical Markup Language as a means of standardising the representation of Auto-ID information has been the subject of several white papers and reports (see Brock, 2001a, Floerkmeier and Koh, 2002). Most of the focus to date in this work has been on the development of standards for parameter and data representation in so called Core Elements (telemetry, product data) and Extension Elements (e.g. data relating to business transactions).

In this section we examine the role of rules and recipes in an Auto-ID information environment – noting particularly that Auto-ID information is by definition **product oriented** which includes **batches** of products or **orders** of products as well as single items.

3.2. Recipes and a Single Identification Standard

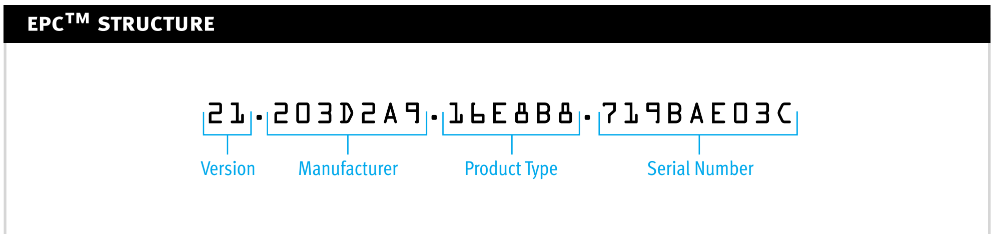
The availability of a single identification standard for Auto-ID systems – the EPC™ (see Brock, 2001) – will ultimately lead to a common, product oriented data structure along the entire supply chain (see Zaharudin et al, 2002). Numerous data base repositories will hold information common to that single EPC™ which can be accessed in a similar manner to current World Wide Web access. The unique ID has at least three implications for recipes (and also for rules):

- Dynamic access to recipe information at any point where an item is within the range of a networked reader. Each **type of product** will have a specific recipe at each stage of its life cycle which may or may not be common to other similar product types. Dynamic access is critical to increasing the flexibility of mobility of products through the supply chain which has numerous benefits.

- Recipes can be partially written in terms of EPC™ information. Referring to Figure 3, Manufacturer and Product Type information can be used to specify raw materials or components used in different operations.

- Each individual item will execute a recipe and will maintain a history record of data collected during that execution. This execution data is item specific and is most sensibly aligned with the EPC™ for the item. In particular, following the point above, raw materials or components used in the execution of a recipe can be completely specified by their unique EPCs™. We will provide an example of this in the next section.

Figure 3



In Figure 4 we provide the example of a simple recipe which describes the assembly instructions for the Gillette Gift Boxes assembled in the University of Cambridge demonstration environment (Hodges et al, 2002). The EPC™ for the overall gift box is provided first then the product-level EPCs™ for three items to be included in the assembly are included below that. In the latter case, the unique serial number is suppressed because at the recipe stage there is no requirement for a unique item to be packaged providing it conforms to the appropriate product type.

Figure 4: Simple Recipe Using EPCs™ for Raw Material Specification

```
<?xml version="1.0" ?>
<node epc="0000000A1000001000000DDD">
<future>
<owner name="retailer">tesco</owner>
<desc>gillete gift pack for tesco summer sale</desc>
<node epc="00000000100000A000000000" />
<node epc="00000000100000B000000000" />
<node epc="00000000100000D000000000" />
</future>

<present>
<owner name="manufacturer">gillette</owner>

<desc>gillette giftbox configuration type A</desc>
</present>
</node>
```

3.3. Recipes and Product Data

As discussed above, the Physical Mark Up Language is being developed as a means of representing item level information associated with a specific instance of a product. Referring to the detailed example of a **product recipe portfolio** in Appendix 1 it is clear that both simple rules and recipes associated with product types will need to draw extensively on data relating to the product for information such as

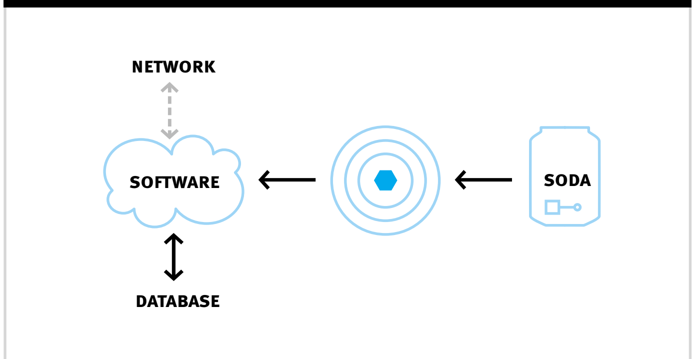
- Material parameters (e.g. current dimensions, composition)
- Telemetry data (e.g. information about the exact location of an item to be processed)
- Historical data that might influence the execution of a rule or a recipe (e.g. temperature history of the product influencing its shelf life and hence retail priorities)
- Customer order/transaction information (e.g. the country of destination of a product which might alter packing instructions)

3.4. Recipes and "Intelligent Products"

In several recent papers (McFarlane, 2002, Zaharudin et al, 2002) the concept of an intelligent product has been introduced, which draws heavily on the Auto-ID approach of linking a physical product to networked information. (See Figure 5). An intelligent product is defined in the following way:

Figure 5

INTELLIGENT PRODUCT MODEL



An **intelligent product** is a physical and information based representation of an item which:

1) possesses a unique identification
2) is capable of communicating effectively with its environment
3) can retain or store data about itself
4) deploys a language to display its features, production requirements etc.
5) is capable of participating in or making decisions relevant to its own destiny

As discussed in (McFarlane, 2002, Zaharudin et al, 2002) this implies that a product is not simply a passive item linked to information relating to it, but can in fact be actively involved in decisions relating to the way in which it moves through the supply chain. The ability to represent product requirements in terms of rules and recipes is critical for the development of approaches in which the product **drives** operations applied to it.

4. HOW WILL RECIPES BE USED IN AUTO-ID ENVIRONMENTS?

4.1. Introduction

We now consider recipe presentation and language requirements in line with Auto-ID standards. One approach is to incorporate the product recipes within a PML datafile.

We note from Section 1, that the basic information types required to specify a recipe are:

- **Recipe Administration**
- **Geometrical Representations**
- **Rules/Constraints**
- **Processing Steps**

In this section we will discuss the way in which Auto-ID information systems can be adapted to accommodate recipe information.

4.2. Recipes and PML

We will predominantly consider the role of PML in accommodating recipe information and in order to do so we make some brief notes about the basis for PML.

4.2.1. PML Background

PML is intended to be a common "language" for describing physical objects, processes, and environments.[1] The range of tags available in the current version of PML is focused towards basic product administration. They allow for descriptions of hierarchical product composition, product ownership and locations. They also allow for measurements such as temperature and mass to be associated with products with respect to time or location.

PML is a mark-up language that utilises the namespace feature of XML. This means that the XML syntax is used to define PML. All PML elements, such as measurement, location, ownership, are defined in a DTD (Document Type Definition). This is a document used in XML related languages to define all the elements and the hierarchical relationships between them. A reference to this document is made in all PML data.

PML is intended to be a mark-up language to describe the form, fit and function of products. As it stands it is biased towards a language for the tracking and identification of components. This is because it has been developed specifically for the first phase rollout of the Auto-ID project. There are currently no product description or product processing elements, although these are under consideration.

4.2.2. Recipe Specifications in PML

Based on the above comments, to develop recipe specifications in line with PML as it currently stands means to use PML tags wherever possible and maintain XML compliance elsewhere.

Some of the information required within a recipe can be classified into PML extensions. PML Extensions are used to integrate information that is not generated by the Auto-ID infrastructure and is aggregated from other sources[2]. (An example is the PML Commerce Extension, which involves process standards to enable transactions within and between organisations to take place.) PML Core, on the other hand, provides the common standardised vocabulary to distribute information directly captured from the Auto-ID infrastructure, e.g. composition, location.

[1] http://www.mit.edu/~tmilne/pml/index.html

[2] Physical Mark-Up Language Update, C. Floerkemeier & R. Koh, June 2002

Recipe Administration

There may be a case though for an additional development of the PML core standard to cater for the requirements of Recipe administration. A future situation may involve the creation of recipe administration details within the Auto-ID infrastructure. Tags that have been identified as a requirement in the case study examples in Carr (2002) (and which also feature in the ISA S88 standard – see) are as follows:

- **Recipe Owner**
- **Version number**
- **Approved by**

These tags, together with the existing PML tags listed below,

- **Date**
- **Owner**
- **EPC™**
- **Datum**

provide the information necessary to fulfil the requirements for recipe administration. (Carr 2002).

Product Representations

It is likely that the specific tags used to describe product components and ingredients will come from the PML extensions. These will utilise, where possible, existing industrial standards such as the RosettaNet technical dictionary. (See Floerkmeier and Koh, 2002).

Rules and Constraints/Processing Steps

There is no means of describing processes or defining rules in PML, which are essential parts of most recipes. It is recommended that rather than inventing a new mark up language to support rules and constraints in Auto-ID environments, that other, established standards, are utilised for descriptions of rules and production processes. This issue is a matter of ongoing investigation, and several options have been examined.

In the area of processing steps, several industrial standards exist (see next section) and these will be examined to determine if they may be suitable for adoption or as guidelines in this context. In the area of rules and constraints, there is no existing industrial standard, and it is worth commenting on this issue in a little more detail:

[2] http://www.dfki.uni-kl.de/ruleml/

One possibility being considered for the specification of rules and constraints is the integration of so called RuleML methods into the recipe description, RuleML is a mark-up language for the standardisation of rules[3]. The RuleML standardization initiative was started in August 2000. Experts in rule theory and technology have subsequently joined the RuleML steering committee, and a group of RuleML participants has been formed to get the expertise and involvement of a large number of academic researchers and industry experts from rule software vendors. The strategy of the RuleML working group is to provide a standard mark-up for rules that can run anywhere in any environment. The group has expressed much interest in STEP and other initiatives outside of the business process arena.

RuleML works with 3 different forms of rules (see Table 2). According to the RuleML taxonomy, the main rule types to feature in recipes are reaction rules and constraints. Reaction rules are concerned with the invocation of actions in response to events. An integrity constraint is an assertion that must be satisfied. There are state constraints and process (or behavior) constraints. State constraints must hold at any point in time. Process constraints restrict the admissible transitions from one state of the system to another.

Table 2

RULE TYPES IN RULEML

REACTION RULES	DERIVATION RULES	CONSTRAINTS
Event Condition Action Effect e.g. Production rules	Implications e.g. If '….' is TRUE then '….' must be TRUE	An assertion that must be satisfied State constraints Process Constraints

The current version of RuleML (version 0.8) does not specify the mark-up of reaction rules. However the next version, RuleML 0.85, due for release in 2002 will define a standard for reaction rules.

4.3. Summary

It appears logical that PML be considered the primary focus for the embedding of recipe information within Auto-ID systems environments. At present it appears that PML Core can satisfy Recipe Administration requirements, PML Extensions can be developed to handle product specifications but that an additional standard need be sought to cover rules and processing steps. RuleML is one candidate being considered.

5. KEY ISSUES

We conclude this white paper by discussing some additional key issues relevant to rules and recipes and their inclusion in an Auto-ID Systems environment. We note that a number of these issues are discussed in greater detail in Carr (2002)

5.1. Recipe Classification

A major constraint in considering a single standard for the inclusion of recipe information into Auto-ID systems is the sheer diversity of the nature of a recipe which can range from a simple household item assembly whose "make recipe" is (almost) completely defined by the nature of its constituent components to a complex chemical subject to constraints both in terms of final product tolerances and also processing requirements. Clearly, quite detailed processing steps and rules are required for such production. This range of recipes is illustrated in Table 3 for the manufacturing section of a product recipe.

Table 3

IMPACT OF PRODUCT AND PROCESS CONSTRAINTS ON RECIPE		
PRODUCT CONSTRAINTS	**Few Constraints**	**Many Constraints**
Loosely Constrained	**[e.g. Basic Assembly, High Vol Discrete Manufacturing]** Recipe Type: Geometric representation/representation of final product e.g. CAD files	**[e.g. Job Shop Manufacture]** Recipe Type: Rules, options and parameters with processing steps made clear
Highly Constrained	**[e.g. Complex Assembly, Flexible Production]** Recipe Type: Geometric representation with order constraints and recommended route	**[e.g. Chemical Batch Processing]** Recipe Type: Rules, options and parameters with clearly defined equipment independent processing steps

5.2. Industrial Recipe Standards

This white paper has not focussed in any detail on existing industrial standards which cover all or part of a product recipe. This is clearly a critical issue for further work, in which the interface between an Auto-ID based recipe specification and existing standards is examined. In this section we simply summarise these relevant industry standards (and de facto standards) and identify their characteristics relevant to our recipe requirements specification. We note that a detailed description of these standards and links to further information is provided in Carr (2002), where standard status, usage, organisational data and web links are provided. The main relevant standards are:

- **EAN-UCC** – A series of standards designed to improve supply chain management. Standards for unique product numbering system, data carriers and e-commerce.

- **GenCAM** – Standard to describe printed circuit boards, and electronic assemblies. This standard is mainly used to define board layout and connectivity.

- **IGES** – Initial Graphics Exchange Specification

- **ISA-S88** – A batch control standard. The standard defines recipes for use within the batch processing industry. The 'general recipe' described in the standard is independent of resources and fully describes the processing stages of a product. The standard was developed for use in the batch processing industry and is not used in the manufacture of discrete components.

- **RosettaNet** – Provides a common language for defining the form, fit and function of any product or service.

- **SRFF** – Surface Mount Equipment Manufacturers Association SMEMA Standard Recipe File Format (SRFF) – used with GenCAM standard

- **STEP** – Standards to define product design in terms of geometry, topology, tolerances, relationships, attributes, assemblies, configuration and more. Contains a series of Application Protocols (APs) to describe various different types of product-data applications. The STEP format allows the transfer of product/component design information between organisations and systems. Most major CAD systems have a mapping to the STEP format.

– **UCCNet** – Standards to allow organisations to communicate product information in a format agreed upon by industry peers. Uses UCC standards to synchronise item information. Provision of a trading community with synchronised data.

– **UDDI** – Project to create a platform-independent, open framework for describing services, discovering businesses, and integrating business services using the Internet, as well as an operational registry that is available today.

– **UDEX** – Global standards organised into a hierarchical structure of Department, Product Category, Product Cohorts and Product Attributes. The aim is to represent both the physical properties and also the marketing claims.

Table 4 gives a summary of the degree to which each of the above industry standards address the required recipe information types introduced in Section 1.1, namely

– **Recipe Administration**
– **Geometrical Representations**
– **Rules/Constraints**
– **Processing Steps**

We note in each case that the examination of the standard has been cursory rather than rigorous and further investigation is required.

Table 4

STANDARDS AND RECIPE REQUIREMENTS

Standard	Geometric Definition	Processing Steps	Product Description	Lifecycle Requirements
EAN-UCC			●	
GenCAM	●			
IGES	●			
ISA-S88		●		
RosettaNet		(in progress)	●	(technical support)
SRFF		●		
STEP	●	●	●	(in progress)
UCCNet			●	
UDDI			●	
UDEX			●	

It is clear that no one standard meets all of the requirements for full recipe description, and in particular no standard currently addresses the issue of rules and constraints at all. Importantly, we should also note that many of these standards only refer to the production section of the product life cycle and are hence further limited in their applicability. Finally, we note that most of the standards have been developed with a particular industrial sector in mind and that few address cross-sectoral issues in any real sense.

The most important standards identified from an Auto-ID perspective were STEP (ISO 10303), ISA-S88, GenCAM (IPC 2511), SRFF (IPC 2531) and RosettaNet standards. (See Carr (2000) for further details).

Table 5 then identifies the way in which these standards can help in addressing the full level of recipe information required within an Auto-ID context. Here we have included RuleML as a possible means for addressing the Rules/Constraints issue.

Table 5

STANDARDS TO REPRESENT INFORMATION WITHIN RECIPES	
TYPE OF INFORMATION	**USEFUL STANDARDS**
Recipe Administration	PML
Geometrical Representations	STEP, GenCAM
Rules/Constraints	RuleML
Processing Steps, BatchML RosettaNet*	SRFF

All of the standards recommended in Table 5 are XML compliant or have mappings to XML. This is a requirement for the representation of recipes as mentioned in section 8.2. In particular, the scope of RosettaNet makes it very important for the recipe project and the standards relating to manufacture should be monitored closely. The technical dictionary may provide a basis for describing product inputs in recipes, within the RosettaNet industries.

6. CONCLUSIONS AND NEXT STEPS

6.1. Conclusions

This report has introduced the concept of a product recipe in the context of an Auto-ID based information environment. The recipe is a means of encapsulating product requirements in a compact, portable manner and is highly compatible with the Auto-ID information model. We have differentiated explicitly between the idea of a recipe which is product oriented and independent of the resources that act on it and the idea of a machine instruction which is resource oriented and entirely specific to a class of equipment type (see also below).

A recipe has been defined to have four main sections

- **Recipe Administration**
- **Geometrical Representations**
- **Rules/Constraints**
- **Processing Steps**

and initial directions for addressing these have been provided. The areas of Processing Steps and Rules/Constraints are likely to require addition developments beyond those currently envisaged within the PML developments at the Auto-ID Centre.

6.2. Next Steps

The next steps in this area are as follows:

– Further investigation into the scope and applicability of industrial standards, and a detailed investigation into rule based approaches such as RuleML.

– Develop trials and demonstrations for the direct integration of recipe information into Auto-ID environments in order to demonstrate its utility. This is likely to be initially within the Cambridge demonstration environment but an industrial trial should also be planned.

– Produce an Auto-ID Recipe Specification – this is likely to be an adjunct to the PML specification under development.

– Develop automated methods for converting (product) recipes to machine instructions – this would appear to be critical for the integrated deployment of recipe information into automated Auto-ID based environments along the supply chain.

7. REFERENCES

1. **ANSI (1995), ANSI/ISA-88.01-1995 – Batch Control Part 1: Models and Terminology.**
Auto-ID Technical Memo, 2002

2. **H. Boley, (2001) "A Web Data Model Unifying XML and RDF".**
Available at http://www.dfki.uni-kl.de/~boley/xmlrdf.html

3. **T. Bray & D. Brickley, (2001) "What is RDF".**
Available at http://www.xml.com/pub/a/2001/01/24/rdf.html

4. **D. Brock, (2001a) "PML Specifications".**
Auto-ID White Paper MIT-AUTOID-WH-00?, 2001.
Available at http://www.autoidcenter.org/research.asp

5. **D. Brock, (2001), "The Electronic Product Code™ (EPC™) – A Naming Scheme For Physical Objects".**
Auto-ID White Paper MIT-AUTOID-WH-002, 2001.
Available at http://www.autoidcenter.org/research.asp

6. **J. Carr, (2002) "Product Recipes Project, MET 2 Long Project".**
Institute for Manufacturing/Auto-ID Centre, Cambridge, UK

7. **C. Floerkemeier & R. Koh (2002), "Physical Mark-Up Language Update".**
Goldstein, (1999) "A Standards-Based Approach To Integrating Information Across The Electronics Manufacturing Supply Network",
Available at http://www.nemi.org/Newsroom/Articles/ECWC8.pdf

8. **S. Hodges, A. Thorne, A. Garcia, J. Chirn, M. Harrison & D. McFarlane, (2002)**
"Auto-ID Based Control Demonstration. Phase 1: Pick and Place Packing with Conventional Control",
Auto-ID White Paper, 2002,
http://www.autoidcenter.org/research.asp

9. **IPC (1999), IPC-2531, SMEMA Standard Recipe File Format Specification.**
ISO (2001), ISO 10303 Step Application Handbook Version 2

10. **Wagner (2002) "How To Design a General Rule Markup Language?". 2002**

11. **A. Zaharudin, C. Wong , V. Agarwal, D. McFarlane, R. Koh & Y. Kang, (2002),**
"The Intelligent Product Driven Supply Chain"
Auto-ID White Paper CAM-AUTOID-WH-005, 2002.
Available at http://www.autoidcenter.org/research.asp

APPENDIX 1: SAMPLE RECIPE REQUIREMENTS FOR A POT OF SPAGHETTI SAUCE

The material in this appendix is extracted from Carr (2002) and is a preliminary investigation into the establishing an example of the complete requirements for a product recipe. The format used here is one which is readable and understandable by humans. Consideration of a format that is interpretable by software or machines would occur at a later stage when the basic rules of structure have been investigated and standards for representation established.

The overall recipe for a product used here resembles a portfolio of **sub-recipes** reflecting different stages in the lifecycle of the product. These **sub-recipes** are as follows

- **General Recipe Admin**
- **Manufacture**
- **Distribution**
- **Retail**
- **Consumer**
- **End-of-Life**

The example provided here serves to indicate the type of information required within a recipe portfolio for a pot of pasta sauce. (In Carr, 2002, an additional example for a desktop computer is included by way of contrast.) This example was produced to highlight some of the requirements for recipes. It is not intended to be the definitive recipe for a product. It is fairly simplistic but is useful in providing a way to think about recipe content. Wherever feasible the recipe has been populated with sample data.

A1. General Recipe Admin

Recipe Administration
This will consist of recipe id, version number, originator, date of issue, owner, approvals etc. This should occur for each sub-recipe within the recipe portfolio.

```
Recipe ID:          Spaghetti Sauce – Manufacture
Version no.:        1.3
Originator:         Golmio Sauces
Date of Issue:      05/05/2002
Owner:              Golmio Sauces
Approvals:          T.A. Stee 02/05/2002, T.O. Martoe 03/05/2002
```

A2. Manufacture

Inputs
The raw material, subassembly and component inputs for the entire product, with their associated id codes. This will most likely contain the relative quantities required which may be absolute values or equations based upon other parameters (e.g. batch size, other inputs).

Reference may be made to preferred or allowable suppliers.

Tomatoes [supplied]	150.0g
Tomato puree type A [supplied]	10.0g
Onions [supplied]	10.0g
Garlic [supplied]	0.9g
Basil [supplied]	0.5g
Sugar [supplied]	10.0g
Preservatives [supplied]	0.5g
Pot [supplied]	1
Label	1
Lid [supplied]	1

Allowable substitute inputs

This will include the entire range of inputs and the associated modifications to input ratios and quantity formulas.

Alternative combination:

Tomatoes	151.0g
Tomato puree type B	8.0g
Onions	10.0g
Garlic	0.9g
Basil	0.5g
Sugar	11.0g
Preservatives	0.5g
Pot	1
Label	1
Lid	1

States

A description of the major states through which the product must pass until it is transformed to its final state.

```
Separate raw materials
Prepared raw materials
Materials mixed in the required quantities
Mixture heated to required temperature
Mixture cooled to required temperature
Mixture packaged
```

Actions

Each change of state will require one or more actions to occur. These actions may be a requirement to move to the next state (by helping to reach a trigger point). The actions that are aligned vertically may be performed in parallel if the resources are available. The actions that are indented from the previous action must be performed after that action.

```
Separate raw materials:
          Wash tomatoes
                    Chop tomatoes
                              Wash to deseed
          Peel onions
                    Chop onions
          Peel garlic to separate cloves
                    Peel cloves
                              Crush cloves
          Wash basil
                    Separate leaves from main stalk
                              Chop leaves
Prepared raw materials:
          Weigh QUANTITY A of tomatoes
          Weigh QUANTITY B of tomato puree
          Weigh QUANTITY C of onions
          Weigh QUANTITY D of garlic
          Weigh QUANTITY E of basil
          Weigh QUANTITY F of sugar
          Measure QUANTITY G of Preservatives
                    Stir for at least TIME A at a minimum STIRRATE B
Materials mixed in the required quantities:
          Apply distributed heat to the mixture to provide a uniform
          heating rate
                    Stop heating when TEMPERATURE X is reached.
Mixture heated to required temperature:
          Cool the mixture at a uniform rate at atmospheric pressure
                    Progress to the next stage can occurs when
                    mixture reaches TEMPERATURE Y
Mixture cooled to required temperature:
          Weigh QUANTITY Z into pot
                    Apply lid to pot with a maximum TORQUE A
Mixture packaged
```

Triggers/Parameters

A trigger is required to progress from one state to another, the trigger may consist of set points, comparison values or values used in conditional logic. The trigger may be a formula containing other variables within this section, or externally referenced variables or values.

```
Separate raw materials:
          All relevant ingredients washed, peeled and chopped
Prepared raw materials:
          QUANTITY A
          QUANTITY B
          QUANTITY C
          QUANTITY D
          QUANTITY E
          QUANTITY F
          QUANTITY G
          TIME A
          STIRRATE B
```

```
Materials mixed in the required quantities:
          TEMPERATURE X
Mixture heated to required temperature:
          TEMPERATURE Y
Mixture cooled to required temperature:
          QUANTITY Z
          TORQUE A
Mixture packaged
```

Outputs
This section details the output and quality requirements at each stage (e.g. tolerance on the ratio of input ingredients, quantity of product in a pot etc.).

```
Acceptable deviations:
          Input ingredients +/- 0.1%
          Quantity in pot +/- 0.5%
```

Production Equipment Requirements
This section will detail any general requirements such as allowable materials, required processing characteristics, safety requirements, legislation etc.

```
Stainless steel only
```

Packaging Requirements
This section will outline the exact specifications for the required packaging type. This may involve a link to a compatible CAD file to define the exact locations and hierarchies of components. A compatible CAD file may be an output from a mapping to a standard for product data exchange such as a STEP file. Mappings to XML from these files can now be achieved.

The description of the various components of the package should detail the connectivity between the separate parts. A reference point for a coordinate system is required so that the relative positions of all labels can be derived.

```
Pot:      Geometry   (URL reference to data file)

Lid:      Geometry   (URL reference to data file)
          Graphics   (URL reference to image file)

Label:    Geometry   (URL reference to data file)
          Graphics   (URL reference to image file)

Assembly: Connectivity and relative positioning
                     URL reference to data file)
```

Other Production Information
This section will detail the support information not featured elsewhere in the recipe (e.g. regulatory compliance information, materials and process safety information etc.)

A3. Retail and Distribution

Storage

This will detail storage requirements of the product and may involve a range of set points and formulae. An example would be a formula to define the picking priority of a product based on the time to expiry (or display until date).

```
States      Priority code 1
            Priority code 2
            Priority code 3
Actions
            Priority code 1
                        When there is less than TIME X until expiry
                        date update the priority code to Priority code 2
            Priority code 2
                        When there is less than TIME Y until expiry
                        date update the priority code to Priority code 3
            Priority code 3
Triggers/Parameters
            Priority code 1 (Reference to priority semantic set by
            brand owner/manufacturer/retailer)
            Priority code 2
            Priority code 3
            TIME X
            TIME Y
```

Pricing

This may be a set of rules governing the pricing arrangements relating to the product. These rules would be defined by the brand owner, the manufacturer or the retailer depending on the agreement. An example would be a formula to define the adjustment of pricing based on the time to expiry (or display until date).

```
States      Pricing code 1
            Pricing code 2
            Pricing code 3
Actions
            Pricing code 1
                  When there is less than TIME X until expiry date
                  update the Pricing code to Pricing code 2
            Pricing code 2
                  When there is less than TIME Y until expiry date
                  update the Pricing code to Pricing code 3
            Pricing code 3
Triggers/Parameters
            Pricing code 1 (Reference to value set by brand
            owner/manufacturer/retailer)
            Pricing code 2 (Either a reference to an external value or
            a formula) Pricing code 3 (Either a reference to an
            external value or a formula)
            TIME X
            TIME Y
```

Display Until

This would be a set of rules to define the display until date based on brand owner, manufacturer or retailer rules and readings taken during manufacture, distribution and retail (e.g. date may have to be modified due to exposure to temperatures outside of an allowable range.). This may be a simple formula relating to the expiry date.

```
States      Date code 1
            Date code 2
            Date code 3
Actions
            Date code 1
                  If temperature is above TEMPERATURE X but less than
                  TEMPERATURE Y for any of the temperature MEASURES
                  A-Z for between TIME A and TIME B then update to
                  Date code 2
            Date code 2
                  If temperature is above TEMPERATURE X but less than
                  TEMPERATURE Y for any of the temperature MEASURES
                  A-Z for more than TIME B then update to Date code 2
            Date code 3
Triggers/Parameters
            Date code 1 (Reference to a formula or value set by brand
            owner/manufacturer)
            Date code 2
            Date code 3
            TEMPERATURE X
            TEMPERATURE Y
            TIME A
            TIME B
```

Expiry

This would be a set of rules to define the expiry date based on brand owner, manufacturer or retailer rules and readings taken during manufacture, distribution and retail (e.g. date may have to be modified due to exposure to temperatures outside of an allowable range.).

```
States      Date code 1
            Date code 2
            Date code 3
Actions
            Date code 1
                  If temperature is above TEMPERATURE X but less than
                  TEMPERATURE Y for any of the temperature MEASURES
                  A-Z for between TIME A and TIME B then update to
Date code 2
            Date code 2
                  If temperature is above TEMPERATURE X but less than
                  TEMPERATURE Y for any of the temperature MEASURES
                  A-Z for more than TIME B then update to Date code 2
            Date code 3
```

```
Triggers/Parameters
          Date code 1 (Reference to a formula or value set
          by brand owner/manufacturer)
          Date code 2
          Date code 3
          TEMPERATURE X
          TEMPERATURE Y
          TIME A
          TIME B
```

A4. Functional

States

A description of the major states through which the product will pass during use.

```
Uncooked pot of sauce prepared for microwave cooking
Cooked pot of sauce
```

Actions

Each change of state will require one or more actions to occur.

```
Uncooked pot of sauce prepared for microwave cooking
Heat to TEMPERATURE A for TIME X
Cooked pot of sauce
```

Triggers

A trigger is required to progress from one state to another, the trigger may consist of set points, comparison values or values used in conditional logic.

```
TEMPERATURE A
TIME X
```

A5. End-of-Life

In the case of the pot of pasta this section need only contain an identification of the packaging material. It may be a reference to another part of the recipe, which identifies the packaging material.

Designed and produced by Foxner. www.foxner.com