Effective RFID-based object tracking for manufacturing

Brusey, J. and McFarlane, D.C.

Effective RFID-based Object Tracking for Manufacturing

James Brusey[*] Duncan C. McFarlane [†]

19th September 2007

Abstract

Automated Identification and in particular, Radio Frequency Identification (RFID) promises to assist with the automation of mass customised production processes by simplifying the retrieval, tracking and usage of highly specialised components. RFID has long been used to gather a history or trace of object movements, but its use as an integral part of the automated control process is yet to be fully exploited. Such (automated) use places stringent demands on the quality of the sensor data collected and the method used to interpret that data. In particular, this paper focuses on the issue of correctly identifying, tracking and dealing with aggregated objects in customised production with the use of RFID. In particular, this work presents approaches for making best use of RFID data in this context. The presented approach is evaluated in the context of a laboratory manufacturing system that produces customised gift boxes.

[*] Cogent Computing Applied Research Centre, Coventry University, Priory Street, Coventry CV1 5FB, UK

[†] Institute for Manufacturing, Cambridge University Engineering Department, Mill Lane, Cambridge CB2 1RX, UK

1 Introduction

Consumers are increasingly expecting to be able to customise their purchases to suit specific needs [19]. An example would be the purchase of a new computer where there are a number of parameters (memory, hard disk, video card) for the customer to select. This sort of customised manufacture is often referred to as *late-stage customisation* since all of the different options that the customisation provides can be handled during the last phases of the manufacturing process [21].

Given this trend toward more flexible production processes where hundreds of different types of end-product are produced by combining component parts in different ways, automation needs to become more sophisticated. Without having dedicated lines for each product type, and by assuming that the end-product is produced to order rather than to stock, there is the need to be able to rapidly switch between multiple operations. Implicitly, this requires highly flexible machines which must be able to quickly determine the appropriate operation to perform. Achieving full automation in such circumstances remains challenging. However, at least some automation of such built-to-order production is certainly achievable. At Dell Computer's OptiPlex plant, for example, the process of transporting parts around the factory is automated in such a way that each workstation receives only the parts it needs when it needs them [18]. However the final product assembly is still a manual process.

Completely automating such late-stage customisation requires more intelligent automation and better sensory information than have traditionally been available. This is because the decision making in a customisable process does not depend on the mere presence of the object, but on which type of object, and sometimes on the specific identity of that object. For example, computer chassis A will be shipped to customer X, who requires 256Mb of memory, while computer chassis B will be shipped to Y, who requires 1Gb of memory. Thus when a computer chassis arrives at a workstation where memory chips are inserted, chassis A must be treated differently to chassis B.

The general problem of establishing and keeping track of the identity and location of physical objects, such as these chassis, is referred to here as the *tracking problem*.

A system that *tracks* objects must provide, on demand:

1. the *location* of an object (where is the memory chip for chassis *B*?)

2. the *state* of an object (does chassis *A* already have memory installed?)

3. the *identity* of an object at a particular location (which chassis is this?).

For such a system to be usefully integrated in the control of a manufacturing process, the location, state and identity information must be as complete and accurate as possible. Also, the information must be provided in a timely manner to avoid delaying the control process.

There are three ways to derive the necessary tracking information: 1) by current sensor data alone; 2) by a model of the process and some known starting state; or 3) some combination of sensor data and a model.

A sensor driven approach has the disadvantage that many specialised, complex sensors may be required in order that location, state and identity of an object can be derived. There will need to be many of these sensors since prior to many operations in the production process, there will need to be a decision made based on the state of the object (such as, which bore holes have been drilled in an engine block). The sensors are likely to be complex to detect sufficient information accurately. The sensors may also need to be specialised since the aspect of the state required to be known will be specific to the operation about to be performed.

On the other hand, a purely model-based approach may have difficulty dealing with even small deviations between the model and reality. For example, the model of a car plant may say that car *A* is followed by *B* and then *C*. But when *A* is removed from the line to fix a fault, if no sensors tell it otherwise, the model may continue to show the order as A, B, C whereas it has become B, A, C. The consequences are trivial when merely trying to trace production progress but can potentially be catastrophic when they directly affect the control of manufacturing operations.

This paper provides building blocks towards an approach that combines both sophisticated sensors that supply identity information with a model-based approach that allows objects to be tracked when they are out of range of the sensors. In particular, it focuses on the issue of tracking the location not merely of individual objects but aggregates of multiple objects. The following section provides some background on Radio Frequency Identification (RFID), which is used in this paper as the main sensor technology for object tracking. (Note that there are several other possible sensors that could be used to determine the identity of products, such as bar code scanners or vision systems [10].) In this work, the focus is on RFID as this offers some specific advantages but also provides some specific challenges to the implementer. This work attempts to make use of the advantages while attempting to addressing some of the implementation challenges. Section 3 presents an approach to deriving a meaningful model of the structure and contents of aggregated objects using data obtained from RFID sensors. This approach is then evaluated in the context of a laboratory manufacturing system developed at Cambridge University.

2 Background

2.1 RFID primer

RFID or Radio Frequency Identification [6] is a technology originally created for friend or foe transponders in aircraft during the second world war. It involves an asymmetric RF transmitter / receiver pair, where one is, on request, transmitting its identity to the other. The identity transmitter is usually referred to as a *tag*, whereas the identity receiver is known as a *tag reader* or sometimes simply *reader*. As long as the distance between tag and reader is small (within about 0.5 metres for HF or 10 metres for UHF),

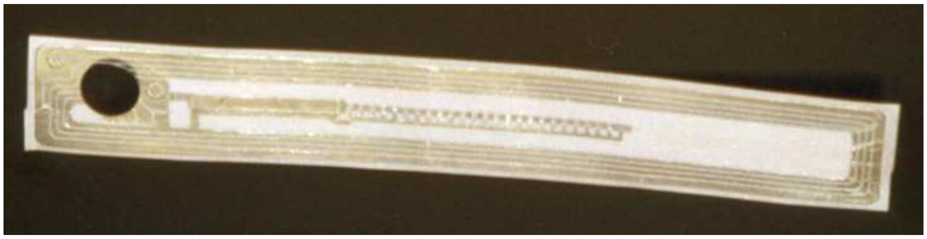

Figure 1: An HF (high frequency) passive RFID tag

it is possible to use a *passive tag*, one that has no battery of its own. Passive tags (such as the one shown in figure 1) operate by absorbing some of the energy in the RF signal transmitted by the reader, and then transmitting back a short message. Key advantages of passive tags are that they are relatively small (around 50mm square and less than 1mm thick), inexpensive, and, due to having no battery, long-lived. This paper deals exclusively with passive tags.

Since passive tags operate by absorbing energy in the RF signal, they tend to operate in bursts rather than continuously. Their response will depend on the local RF signal strength and their orientation relative to the local direction of the RF field. In turn, the local signal strength and direction will depend on what other objects are nearby. Specifically, conductive objects such as metal or liquid filled containers, will distort and deflect the field.

When two or more tags exist in the RF field, they may try to reply at the same time, interfering with each other's response. This effect is referred to as a *tag collision*. Various anti-collision protocols exist for preventing or ameliorating the effect of tag collisions. The simplest of which involve causing the tags to wait for some (possibly random) amount of time before retransmitting (e.g. ALOHA [6]), while more sophisticated algorithms query specific ranges of tags until only a single one replies (e.g. binary search [6]). Even using such sophisticated algorithms, increasing the number of tags in a field will have the effect that any specific tag will be detected less often. In the worst case, and particularly for objects moving quickly past a reader, it is possible for some objects not to be detected at all.

2.2 Related work

RFID technology has been used for a wide range of applications, such as logistics, livestock tracking, and security. Starting in 1999, the Auto-ID Center was formed to develop and promote a common platform for using RFID in the retail sector to track goods as they moved through the supply chain. The central themes of the work were

1. To reduce the cost of RFID tags by minimising the number of bits required to be stored on the tag, and,
2. To combine RFID with computer networking thus providing an "Internet of Things".

Auto-ID later became EPCglobal Inc., and this body has since developed and ratified a number of standards relating to RFID tag data (Tag Data Standard and the Electronic Product Code or EPC), middleware interfaces (Application Level Events), database interfaces (EPC Information Services), and mechanisms for finding those databases given an EPC (Object Naming Service) amongst others. The overall EPCglobal architecture is described in an Architectural Framework Standard [20]. Hodges and McFarlane [10] provide a readable introduction to the main Auto-ID concepts.

More recently, the EU funded project PROMISE [13] has been investigating the use of a combination of active and passive RFID tags on products for item-level Product Lifecycle Management (PLM). PROMISE aims to develop Internet-enabled systems for managing individual products from their initial development and production, through use, service and maintenance, through to reuse and recycling at the end of their life. In comparison with Auto-ID, PROMISE focuses on the whole life of the product rather than just that prior to sale and also tends towards more data (such as service history) being stored on the tag, rather than only in back-end databases.

Product-centric approaches, based on RFID technology, have been proposed by Kärkkäinen et al., [12]. In such an approach, the product is logically considered to be acting like an agent—asking various resources for assembly, delivery, or maintenance, as required. This is also termed "inside-out" control, since the control logic is written from the product's perspective, rather than the machine. In principle, this approach simplifies the development of new control logic for the creation of different versions of a product. The approach does not address encapsulation of the automated machinery (such as lathes, drill-presses, robots, and conveyor belts). Without such encapsulation, flexible control logic for the product would need to encode not merely what operation needs to be done but also how to do that operation with each capable machine.

Some possible approaches to encapsulating automation resources lie in the extensive holonic manufacturing literature. McFarlane and Bussmann [15] provide a useful summary. In the holonic view, the conventional centralised planning, scheduling and execution system is broken up into a number of relatively independent "holons", one per resource or product. These holons interact by communicating their requirements, possibly by using some form of auction to find a good fit of product to resource time slot.

Another form of tracking that makes use of RFID has been developed by Hahnel et al., [9]. They have implemented a variant of Monte Carlo Markov Localisation to make use of RFID tags as landmarks for a mobile robot. Their approach addressed a problem that is a corollary of the one examined here; rather than tracking a moving tag with fixed readers, Hahnel et al. tracked a pair of mobile readers given fixed tags.

The use of Petri nets in modelling manufacturing processes is well established [4]. They are implicitly used to track the movement of parts for control purposes by supervisory Petri net control approaches [3].

These related activities form a background to the RFID based automation development at Cambridge, which is described next.

2.3 Previous developments

The system described in this paper builds on two prior phases of development at Cambridge. The first phase was based on a reactive architecture, while the second phase made use of agent-based approaches.

Hodges et al., [11] developed one of the first laboratory manufacturing systems at Cambridge to make use of RFID in a customised manufacturing process. This system made use of a Fanuc M6i 6-DOF robot, two Montech monorail track arranged in two loops, four small Checkpoint RFID readers, four large Checkpoint RFID readers. and four custom developed hoppers. The aim of the system was to flexibly custom-build Gillette$^{\text{TM}}$ gift boxes from a variety of component parts.

In the phase one approach, the system worked as follows. First, shuttles carrying two items each move around a closed loop (loop *A*), pass by a large reader, thus registering their presence and identifying the type of item (gel, razor, foam or deodorant), and come to a stop at a docking station. This arrival triggers the robot to remove both items from the shuttle and place them in one of the four hoppers according to type. At the base of each hopper, a small RFID reader checks the type of the item and registers its presence. The antenna of the small RFID reader was de-tuned slightly so that only the item at the bottom of the hopper is inspected. Independently of this process, shuttles carrying empty boxes move around another closed loop (loop *B*), pass by a large reader and come to a stop at a docking station. This triggers the robot, assuming that it is not busy, to fill the empty gift box with items from the hopper. As a special case, if the item needed to fill the gift box is available on a shuttle on the near-side loop, then this item is transferred by the robot directly.

The phase one approach was largely reactive. The RFID message indicating shuttle arrival triggered the action of the robot to start. This design meant that the software was quite simple but also reliable. Note that both items on each shuttle in loop *A* had to match; it was not considered possible to accurately detect the order of items from the

RFID data.

The second phase of development expanded on the phase one system in two ways [1, 7, 5]. First, it added flexible routing so that all of the monorail track was interconnected and thus finished products and raw materials were intermingled into the same flow. Second, it introduced a multi-agent-based approach, making use of an agent software development tool called JACK$^{\mathrm{TM}}$.

When extending the phase one system to allow products to flow into and out of each manufacturing cell—essentially dealing with routing of parts to appropriate destinations—it was discovered that some knowledge of the state of the system was required to avoid deadlocks. Also, as processing times extended, more failures were traced to unreliability in the RFID sensory data. Two types of errors in interpreting RFID tag reads were identified: *false negatives* where a tagged object is in range but not detected, and *false positives* where a tagged object is outside the expected range but is still detected [2]. In that work, a simple filter was suggested. Further experimental results have been presented by Floerkemeier et al., [8], who have extended the approach to use Bayesian techniques.

The phase two development suffered from a number of design flaws, as described by Evertsz et al., [5]. The use of multiple agents did not lead to a more robust or flexible system. This is not to say that the multi-agent approach is flawed, but merely that it does not necessarily prevent poor design. Furthermore, the complexity of the system had increased dramatically from phase one. It was not clear that such a system would be readily accepted by the manufacturing industry given the apparent cost of developing and maintaining it.

Based on the experience gained with developing the phase two system, a third system was developed. The tracking mechanism developed for the third phase is described and evaluated in the next section and is the main development presented in this paper.

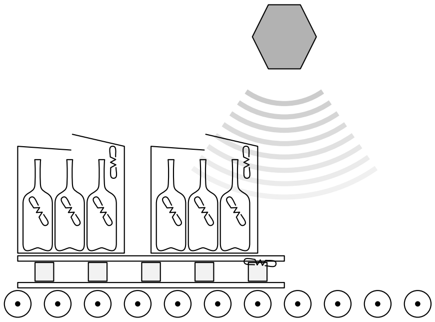

Figure 2: An example aggregate on a conveyor that consists of a pallet with two boxes each containing three bottles.

3 RFID for automated object tracking

In this section, a generic approach is presented to integrate RFID sensor data gathered over time together with a representation of the state of a manufacturing system and with a model of how that state is changed. The aim of this approach is to enhance the accuracy of the identity information and thus improve the robustness of the manufacturing system. It relies on the fact that parts are not always seen in isolation, but often travel together. A common example is that of pallets and cases. Two cases on the same pallet will tend to both be detected by RFID sensors at around the same time. Similarly, the pallet will be detected along with the two cases. All together they form an *aggregate*. Aggregated objects provide an opportunity to improve the reliability of RFID information.

When considering aggregates, such as the one illustrated in figure 2, structure plays some role. For example, it is easier to remove a case from a pallet than to remove the pallet from underneath several cases. Typically this structure is hierarchical. A pallet may contain several cases, each of which may contain some bottles. When a case is removed from the pallet, those bottles that were in the case will move with it.

In general, to understand how objects and object containers move from one location to another, some form of model is required.

This section is organised as follows: First, a basic mechanism for detecting aggregates from RFID sensor data is described. This is then extended to attempt to support inferring whether one object is contained within another. Next, the inference made about the tagged objects must be stored in some form of internal representation and must also be updated according to a transition model when control actions are taken. Finally, the rule-based mechanism used to sequence actions is briefly described.

3.1 Discovering aggregates

In this paper, a time-based approach to determining aggregates is proposed. This approach relies on constraining the flow of each aggregate as it moves past the RFID tag reader. Specifically, there must be a delay both before and after each aggregate is detected by the reader where no tags are detected. In addition, while the aggregate is "seen", the associated tag read events should not be separated by too much of a delay.

Define a string of tag read events occurring at a particular tag reader r as

$$s(r) = (e_1, t_1), (e_2, t_2), \ldots, (e_n, t_n),$$

where e_k is a tag read event that occurred at time t_k. This string is ordered by time such that if $a < b$ then $t_a \leq t_b$. Assume that the aggregate moves past the reader over a particular interval of time and that there are no other tagged objects within the read range at the same time as the aggregate. In this case, all the events for the aggregate passing by the reader will be contiguous within $s(r)$. To ensure that the aggregate can be detected unambiguously, the events for the aggregate should be separated from other events by some time period K. Formally, the sequence of read events $\{e_a, \ldots, e_b\}$ belong to a single aggregate if and only if $t_a - t_{a-1} > K$, $t_{b+1} - t_b > K$ and also iff $t_{i+1} - t_i \leq K$ for all $a \leq i < b$.

Laboratory obtained RFID data for a single reader is shown in figure 3. In this case, the minimum time between aggregates is about 5 seconds, while the maximum

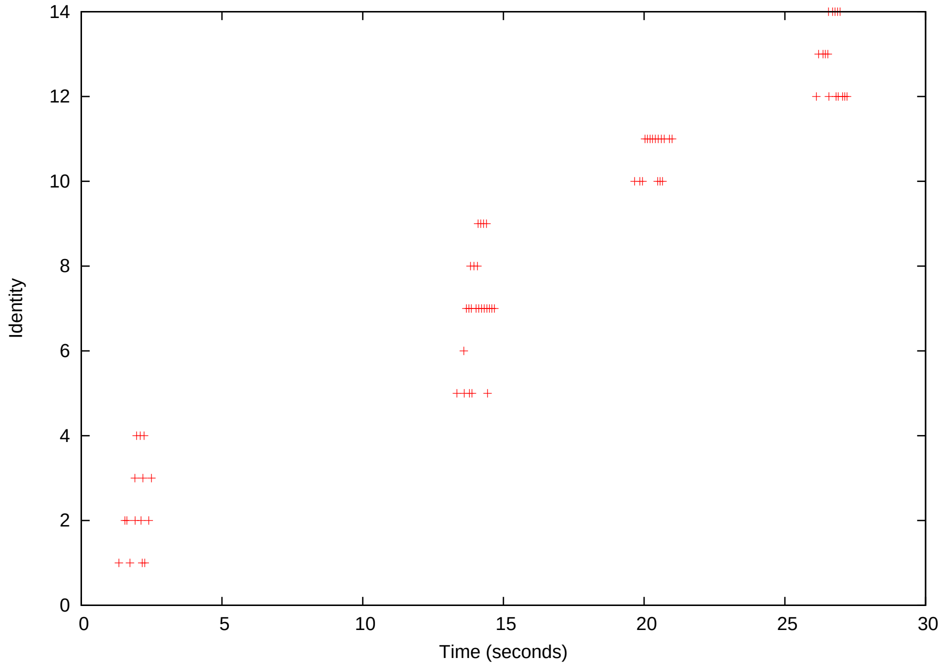

Figure 3: RFID tag reads for a series of four aggregates passing by a reader.

time between events for the same aggregate was about 200 milliseconds.

The choice of the parameter K must be sufficiently large to ensure that a single aggregate is not considered to be two separate objects but at the same time, sufficiently small so that two aggregates arriving one after the other are not considered as though they were a single object. For example, K should be large enough so that if a tag at the leading edge of the aggregate is seen as soon as the object moves into the field followed by a tag at the trailing edge being seen when the aggregate leaves the field, then the aggregate is still seen as a single object. Specifically, for an aggregate of length l_a travelling at a fixed velocity v past a read field of length l_r (as shown in figure 4), then we require that

$$K > (l_a + l_r)/v.$$

In some cases, it may be necessary to constrain the flow of aggregates to ensure that

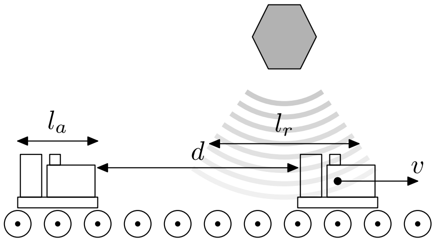

Figure 4: Example of two aggregates moving along a conveyor, past an RFID antenna. The length of the aggregate l_a, its velocity v, and the size of the antenna's field l_r are related to the minimum allowable distance d between aggregates.

each arrives at the reader a small time after the prior aggregate has moved out of the way. Specifically, let the distance between two aggregates (from trailing edge of the first to leading edge of the next) be d, as shown in figure 4. Then an additional constraint is $K < (d - l_r)/v$ which can be rearranged to give a spacing requirement of

$$d > Kv + l_r.$$

Another issue is that of whether it is allowable for the aggregate to stop near the reader. The main difficulty with this is due to the existence of regions near the reader where a tag can be placed indefinitely without generating a tag read [14]. For the above approach to work, it would be necessary to set K to be at least as large as the maximum time spent stopped.

One reason that it may be necessary to slow down or stop the aggregate as it passes through the read range is to allow all of the tags to be read. If an aggregate involves many sub-components, and at least some components are tagged, then multiple tags will be in range of the tag reader simultaneously. Obviously if all tags attempt to respond simultaneously then their signals will interfere. For this reason, tag readers and tags typically employ some form of anti-collision protocol, such as ALOHA or binary search [6]. ALOHA is one of the simplest mechanisms and relies on each RFID tag only responding intermittently thus reducing the probability of a tag collision. However as the number of tags increase, the length of time needed to be reasonably confident that all tags have been detected also increases. For a 99.9% confidence level, Finkenzeller [6] suggests that for HF tags, 0.5 seconds is required to see 2 tags, whereas for 8 tags, 2.7 seconds is required. Different anti-collision protocols have different characteristics but all require longer periods to recognise larger numbers of tags.

Given that tag collisions and other environmental factors may result in some tags in the aggregate being missed, tracking the movement of the aggregate, rather than the individual object, allows such missed tag reads to be inferred. This is a key benefit of this approach.

Once aggregates have been discovered, prior knowledge about the characteristics of the tagged objects can help to infer the aggregate's likely structure.

3.2 Inferring containment relationships

When a set of objects form an aggregate, it is usually the case that at least one of the objects acts as a container. For example, a pallet that supports cases can be considered to "contain" those cases, in the sense that if the pallet moves, then so do all of the associated cases. The converse is not necessarily true. Sometimes a case will be removed from a pallet. The notion of containment is naturally hierarchical, and so cases may contain, say, bottles of wine. When the case is removed from the pallet, the bottles contained within that case will move too.

In any given application, there are typically only a few levels of the containment hierarchy, and also only a few ways that containment can occur. To infer the likely containment structure, it is usually sufficient to know the likely containment *level* of each type of object. For example, a pallet might be of level 1, a case of level 2, and a bottle of level 3. Then, in an application where a bottle should never appear on a pallet on its own, the appearance of a single pallet, a single case and a single bottle allows us to infer that the bottle is contained by the case and that the case is contained by a pallet.

When several tagged objects exist at the same level, for example, two cases are detected but only a single bottle, then it is not possible to infer the location of the bottle. However it is possible to say that, in the absence of any other information, that it is equally likely for the bottle to be in either case. This probabilistic representation of the position of the bottle may not be useful immediately, but if subsequently one of the cases is removed, and the pallet subsequently passes by a reader, the absence of the bottle at this stage implies that the bottle is more probably in the case that was removed.

3.3 World state representation

In order to concisely represent the state of a large number of tracked objects, some simplification and approximation is necessary. RFID reads are only detecting objects at a finite set of known locations L.

Based on experience from previous phases, the location of objects was stored internally as a mapping from tagged object to location. This mapping can be expressed as a function $f : I \to (I \cup L) \times \mathbb{Z} \times \mathbb{Z} \times \Re$ where I is the set of identified (tagged) objects, L is the set of locations, $\mathbb{Z}$ is the set of non-negative integers, and $\Re$ is a real number. For a tagged object i, $f(i)$ gives an ordered tuple (l,k,t,w), which includes the location l (which may be another object), an index k, a time-stamp t, and a likelihood estimate (or weight) w. Allowing objects to act as locations is the mechanism for representing containment. This approach means that when the aggregate moves, only the bottom level container location needs to be updated. The index k is used to represent putting multiple objects into a single container at the same level of containment. The time-stamp t keeps the time of the most recent update thus allowing old, out-of-date, RFID data to be discarded. Finally, the likelihood estimate w is used to determine which of several objects is actually at a location based on the relative frequency of receiving RFID reads.

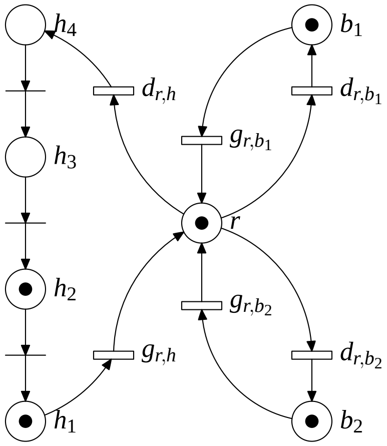

Figure 5: Part of the Petri net transition model.

3.4 Transition model

In its essence, tracking involves detecting when the state (location) of an object changes. Such change occurs in either through explicit control action (such as a robot picking up and moving an object) or implicitly (such as objects falling down a hopper or flowing along a conveyor belt).

To estimate the change in state caused by a control action, some form of model is required. A form of high-level Petri net [16] was used to describe the object tracking problem examined in this paper. This net represented the possible locations for objects as *places* while actions are represented by *transitions*. A *token* in a place represents an object being at a location. Since objects are identified uniquely, each corresponding token has a corresponding identity. A portion of the model used is shown in figure 5. Note that, for conciseness, the token identity is not shown in the diagram. Controlled transitions, shown as boxes in the diagram, are labelled with the corresponding action. Uncontrolled transitions, unlabelled and represented as lines rather than boxes, can occur at any time as long as there is an object at the source place and nothing at the target place. Requiring that the target place be empty is not usual for Petri nets, however it is helpful here since tokens correspond to uniquely identified objects, and their order, for example in the work-in-process buffer, must be preserved in the model. When a transition fires, an object is moved from the transition's input place to its output place. For example, a typical action in an automated assembly system is for a robot r to "grab" an item from a hopper h, and this action is denoted $g_{r,h}$. The places correspond to possible object locations, such as r for the robot, $h_1, \ldots, h_4$ for the four positions in hopper h, and b_1 and b_2 for two locations in a box. There are two types of action shown in the diagram: $g_{r,x}$ being a "grab" from x to r, $d_{r,x}$ being a "drop" from r to x where $x \in \{h_1, \ldots, h_4, b_1, b_2\}$.

A key issue with the development of the transition model was the correct handling of asynchronous updates to the world state from the transition model and RFID sensors. Network and processing delays can mean that the last few tag reads for an object that has just been moved away from a reader arrive after the transition model has updated the location of the object. In early versions of the development of this approach, the model indicated that an object apparently "jumped" back to its previous location. This was clearly not the case but the result of processing old RFID data after updating the object state based on an action. To resolve this, RFID reads events are timestamped at the source and any events older than updates from the transition model are ignored.

Note that actions are not derived from the transition model, but rather come from a reactive rule system. The interaction of RFID sensor data, the transition model, the rule system and the world state representation is shown in figure 6.

Generation of control actions is not performed by the transition model, nor by a planning system based around the transition model. Instead, a reactive rule system is used. Following an approach suggested by Nilsson [17], the rule system is goal-oriented and has a recursive structure. Each rule is a combination of a predicate and an action. The action may either be an individual control action, such as a robotic movement of an item, or a sub-goal, represented as a set of rules. For example, a sub-goal might be to pack a box with a gel, a foam and a razor. This then breaks down into

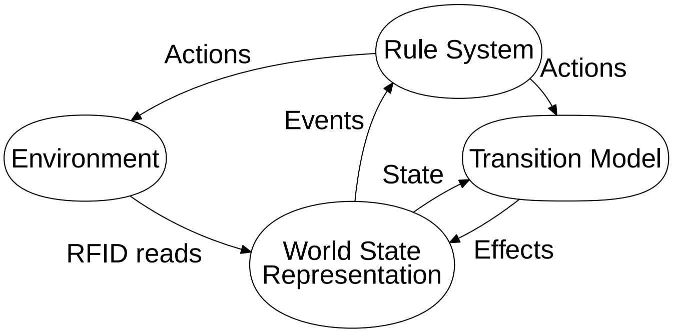

Figure 6: Diagram showing the flow of information within the overall system. An internal world state representation and a transition model are used to track the effects of actions. The world state representation tracks the location of parts first by interpreting RFID tag reads, but also by interpreting the effect of actions on the current state using a transition model.

the rules to move the box to the robot and then to move the individual items into the box. The rules are ordered so that rules about situations close to the goal are presented first, while situations further from the goal are presented later.

An important feature of the rule system architecture is that it reacts to notification of changes to the world state representation rather than sensory signals direct from the environment. This is, in effect, a state based filter and was found to be important in ensuring that the overall system functions reliably.

4 Evaluation

To evaluate the approach to object tracking described in the previous section, it was applied to the Cambridge laboratory manufacturing system mentioned previously. This system packs Gillette$^{\text{TM}}$ gift boxes. As with previous development phases, it packs to order rather than to stock. It extends earlier work by both routing parts and boxes to the appropriate cell and flexibly handling the packing operation of a single box across several cells. It also removes the finished product from a shuttle and puts it into a warehouse. The order can be changed at any stage during production, causing the

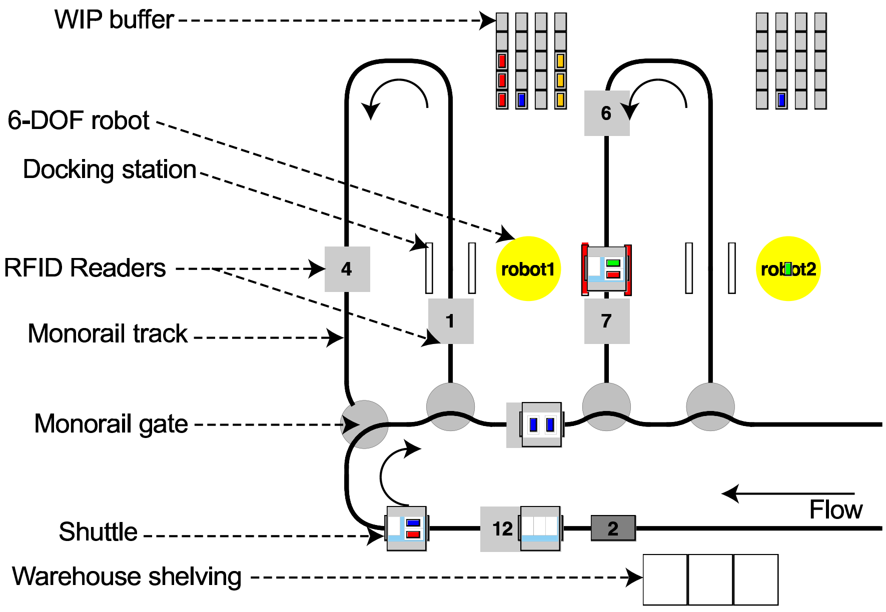

Figure 7: Diagram of the gift-box packing system. Solid arrows show direction of flow of goods on monorail track.

gift box to be repacked in an efficient manner. A schematic diagram of part of the manufacturing system is shown in figure 7.

To allow the location of parts to be identified, RFID tags are attached to the individual items (gel, razors, foam, or deodorant), the boxes, the trays carrying the boxes and the shuttles. RFID readers are positioned at the base of the work-in-process (WIP) stacks (see figure 8) and along the monorail track just prior to the gates and docking stations. Although the original design called for readers prior to every decision point, some readers were able to be disabled, although some slight changes were required to the transition model to cope with this.

The experimental work performed in this paper made use of a simple form of HF tag that uses an anti-collision protocol of transmitting every 100ms +/- 50%, but with each tag factory set to use a slightly different period. The readers used can perform some simple filtering (such as filtering recently seen tag messages), however this was

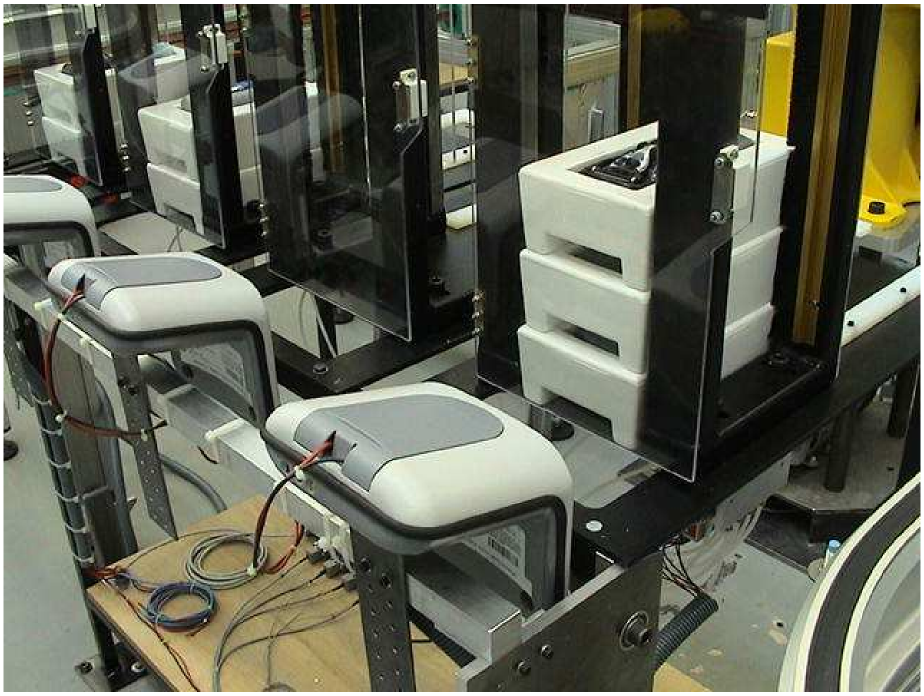

Figure 8: Work in process buffer for packing robot.

turned off.

The aim of the experimental evaluation was to assess how many incorrect RFID-derived inferences could be removed by superimposing the tracking model on top of the sensory data.

4.1 Results

A statistical summary of logs produced by experimental runs of the laboratory manufacturing system is given in table 1. The total running time shown in the table reflects an average of about 30 minutes per run. Experimental runs consist of placing several orders to test the ability of the system to cope with customised demand, and then changing the orders to demonstrate its ability to react to a changed demand. As shown in the table, on average about 2000 object movements per run were detected (via RFID data) or inferred based on explicit control actions or implicit effects (such as items dropping down in a hopper). For each run, about 50 control actions (such as robotic movement of objects or monorail gate switch operations) were taken per run.

It is reasonably common for the system to receive a false positive RFID read in the work-in-process stack (2 per run on average), since the WIP tag readers sometimes read the item second from the base of the stack as well as the item at the base. This leads to two items being considered to be at the base of the stack. Roughly half the time this is resolved when the probability estimate for one of the items reduces below a threshold (a value of 0.2 was used for this threshold) and is discarded. Since only a single item can fit at the base of the stack, the probability of an item being at the base decreases when another item is detected there. In the rest of the cases, the uncertainty was removed after an action was taken to move the item at the base, and subsequently one of the items was detected elsewhere.

The process of forming an aggregate has proven useful in reducing problems caused by false negatives for a shuttle tag. Although the shuttle tag is in close proximity with

Table 1: Accumulated results from 55 demonstration runs

Total running time (minutes)	1 649
Object movements detected or inferred	109 366
Actions taken	2 683
False positives in work-in-process stack	106
False positives pruned by probability threshold	57
False positives pruned after object movement	49
False negatives for shuttle corrected	41
False negatives for shuttle not corrected	2

each reader as it passes by, it is sometimes the case that the shuttle tag is not detected at all. Since seeing the shuttle tag is used to identify the movement of the shuttle and therefore to take actions such as switching a gate, it is critical that the shuttle can be identified. Based on previously gathered aggregate information, it was possible in most cases to correct for the missed tag and thus to keep operating without intervention. The two cases where this was not possible occurred when the shuttle tag was missed on the first occasion that the aggregate was seen.

5 Conclusion

RFID is a mature technology that is currently seeing a rise in prominence, largely due to its increased use in the retail sector. It has been applied to manufacturing, however it is mostly used as a means of establishing the genealogy or history of the end product, rather than as a mechanism to support the automation of customisable production. However increased consumer demand for customisation may drive manufacturers to adopt RFID as a central part of the manufacturing control loop.

Tracking RFID tags in a stateless manner has been demonstrated to be sufficient for many applications, however more sophisticated use of RFID will require the integration of a model-based approach to allow for additional knowledge of object movements to be incorporated. In particular, when RFID is used for automatic control, its reliability can be significantly enhanced by modelling the movement of parts and thus detecting some sensor errors. As a side effect, this can also allow a reduction in the number of RFID readers required and further address the problem of any temporary failure to read tags.

References

[1] James Brusey, Martyn Fletcher, Mark Harrison, Alan Thorne, Steve Hodges, and Duncan McFarlane. Auto-ID based control demonstration - phase 2: Pick and place packing with holonic control. Technical report, Auto-ID Centre, Cambridge University, 2003.

[2] James Brusey, Christian Floerkemeier, Mark Harrison, and Martyn Fletcher. Reasoning about uncertainty in location identification with RFID. In *Proc. IJCAI-03 Workshop on Reasoning with Uncertainty in Robotics*, Acapulco, Mexico, 2003.

[3] Christos G. Cassandras and Stéphane Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.

[4] Frank DiCesare, editor. *Practice of Petri nets in manufacturing*. Chapman & Hall, 1993.

[5] Rick Evertsz, Martyn Fletcher, Richard Jones, Jacquie Jarvis, James Brusey, and Sandy Dance. *Implementing Industrial Multi-Agent Systems using JACK*, chapter Programming Multi-Agent Systems. Springer, 2004. (to appear).

[6] Klaus Finkenzeller. *RFID Handbook: Radio-Frequency Identification Fundamentals and Applications*. John Wiley & Sons, 2000.

[7] Martyn Fletcher and James Brusey. The story of the holonic packing cell. In *Proc. 2nd Intl. Joint Conf. on Autonomous Agents and Multi-Agent Systems (AAMAS-03)*, Melbourne, Australia, July 2003. ACM Press.

[8] Christian Floerkemeier, Matthias Lampe, and Thomas Schoch. The smart box concept for ubiquitous computing environments. In *Proceedings of SOC'2003 (Smart Objects Conference)*, Grenoble, May 2003.

[9] Dirk Hähnel, Wolfram Burgard, Dieter Fox, Ken Fishkin, and Matthai Philipose. Mapping and localization with rfid technology. In *Proc. of the IEEE International Conference on Robotics and Automation (ICRA), 2004*, 2004.

[10] Steve Hodges and Duncan McFarlane. *The Security Economy*, chapter RFID: The concept and the impact. Organisation for Economic Cooperation and Development, 2004. Available online `http://www1.oecd.org/publications/e-book/0304031E.PDF`.

[11] Steve Hodges, Alan Thorne, Andy Garcia, Jin-Lung Chirn, Mark Harrison, and Duncan McFarlane. Auto-id based control demonstration phase 1: Pick and place packing with conventional control. Technical Report CAM-AUTOID-WH-006, Auto-ID Center, 2002. `http://www.autoidlabs.org/whitepapers/CAM-AUTOID-WH-006.pdf`.

[12] Mikko Kärkkäinen, Timo Ala-Risku, and Kary Främling. The product centric approach: a solution to supply network information management problems? *Computers in Industry*, 52(2):147–159, 2003.

[13] Dimitris Kiritsis. Ubiquitous product lifecycle management using product embedded information devices. In *Proc. IMS 2004*, 2004. Available from `http://www.promise-plm.eu` [Online; Accessed 29-January-2006].

[14] Hugo Mallinson. Enhancing identity with location. Master's thesis, University of Cambridge, UK, 2003. `http://www.ifm.eng.cam.ac.uk/automation/publications/`.

[15] D.C. McFarlane and S. Bussmann. Holonic manufacturing control: Rationales, developments and open issues. In S.M. Deen, editor, *Agent-Based Manufacturing*, chapter 13, pages 303–326. Springer-Verlag, Berlin, 2003.

[16] Tadao Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[17] N. Nilsson. Teleo-reactive programs for agent control. *J. Artificial Intelligence Research*, 1:139–158, 1994.

[18] Stacy Perlman. Automate or die. *Business 2.0*, July 10 2001. `http://www.business2.com/b2/web/articles/0,17863,513636,00.html`.

[19] B. J. Pine. *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press.

[20] Ken Traub, Greg Allgair, Henri Barthel, Leo Burnstein, John Garrett, Bernie Hogan, Bryan Rodgrigues, Sanjay Sarma, Johannes Schmidt, Chuck Schramek, Roger Stewart, and KK Suen. The EPCglobal architecture framework, July 2005. Available from `http://www.epcglobalinc.com/standards_technology/Final-epcglobal-arch-20050701.pdf` [Online; accessed 29-January-2006].

[21] Mitchell M. Tseng and Frank T. Piller, editors. *The Customer Centric Enterprise: Advances in Mass Customization and Personalization*. Springer, 2003. Also see `http://www.mass-customization.de`.