REAL-TIME FINE-GRAINED MULTIPLE-TARGET TRACKING ON AN EXTENSIBLE VIRTUAL FAB ARCHITECTURE USING MULTI-AGENTS

Ching-Hu Lu, Chun-Feng Liao, Chao-Lin Wu and Li-Chen Fu*
Department of Computer Science and Information Technology
National Taiwan University
Taipei (106), Taiwan

ABSTRACT

The concept of a “Virtual Fab” (VF) stems from the IC industry and emphasizes the idea of intangible manufacturing service provision. In addition to satisfying customers’ order fulfillment, providing remote service accessibility and real-time data granularity with gratifying performance and flexibility are among major challenges of a VF system. In this paper, we propose an architecture for a VF based on multi-agents that provides fine-grained real-time multiple-target tracking service for both the customers and internal managerial personnel of an IC foundry. Each function module is constructed as an autonomous agent and performs dedicated tasks. To provide flexibility and allow for future enhancements, our proposed extensible architecture also can utilize currently available RFID (Radio Frequency Identification) techniques to cooperate with other tracking sensors for the purpose of improving tracking accuracy and mitigating the power consumption issues inherent in RFID systems. Finally, an integrated simulation which utilizes technologies including RFID, Web Services and embedded-systems has also been conducted to demonstrate the effectiveness of this architecture.

Keywords: Virtual Fab, Multi-Agents, Sensor Fusion, Fine-Grained Multi-Target Tracking, RFID

1. INTRODUCTION

A VF [28,34] means that the service of an IC foundry is so thorough that its customers can virtually take the foundry as their own factory; namely, a VF is a foundry which can be virtually owned, monitored or even controlled by the customers. If such a VF is available, customers would give orders to IC foundries rather than establish self-owned, prohibitive IC factories. The idea of a VF with manufacturing service provision has been proposed by the IC industry as one of the critical aspects for achieving competitiveness [28,33]. Traditional or legacy VF systems can be accessed readily from a web browser. A VF is a convenient tool to allow the customer of an IC manufacturer to monitor the most updated product-related information, such as inventory or WIP (Work In Process [13]) reports. However, constructing a VF is time consuming since the integration of heterogeneous systems and services is needed. In addition to satisfying customers’ order fulfillment, providing remote service accessibility and real-time data granularity with gratifying performance and flexibility are among major design challenges in a VF system.

In order to deal with these challenges, several VF architectures and frameworks have been proposed. For example, [29] proposed a 3-layered VF-enabling framework and fab model, and [6] created a dynamic model for VF services. In this research, we propose a VF architecture consisting of various function modules, each of which is constructed as an autonomous agent and performs specific tasks; that is, each agent is like an independent component that communicates with other agents through standard Web Services mechanisms. As a result, this design facilitates the dynamic deployment of agents and makes the VF systems extensible and scalable such that revision to any individual agent will have minimal impact on others.

Unlike [40], our proposed architecture does not primarily focus on order fulfillment. Here we extend the VF concept further to include mobile-accessible platforms and real-time fine-grained multi-target tracking service provision. In other words, our VF is not restricted solely to the function of customer access on PC-based browsers, but is also available to internal managerial personnel to obtain real-time

* **Corresponding author: lichen@ntu.edu.tw**

information from the shop floor level [7,13] or even the optional Cell-Controller (CC [7]) level. The mobility and flexibility of a VF system is greatly improved when a customer is equipped with a handheld device. In this case, the fab becomes a 24/7 accessible foundry, just as if one were actually physically inside the factory; that is, a VF then becomes several dynamically installed components inside users' hand-held devices and on servers, all of which are tightly integrated with users' daily lives. One could now even send commands to remote equipment for adjusting current operation status instantly. This concept will bring more cost-effective and value-added benefits to the original VF design.

The remainder of this paper is organized as follows: In Section 2, we first present a survey of prior research work in the areas of both VF and indoor tracking. In Section 3, we focus primarily on discussing our proposed architecture and introduce techniques for accomplishing real-time multiple target tracking based on multi-agent mechanism. Before concluding in Section 6, we present an experimental scenario which demonstrates the effectiveness of our proposed architecture.

2. RELATED WORK

2.1 Virtual Fab Survey

The general notion of a VF has been widely accepted as a good vision. Yet, the exact contents and implementation methodology are still indefinite in the IC foundry industry. Most of the current VFs consist of a suite of web-based applications that provide customers with information related to wafer design, engineering, and other financial information.

As an example, consider eFoundry, TSMC's VF [34]. In eFoundry, customers can monitor key information through 24/7 online access to design, engineering and electronic supply chain information, such as wafer process data, purchase orders, WIP reports, shipping notices, and other important logistical information. All information is updated three times a day so that eFoundry can provide abundant decision-making information to customers. Similar to eFoundry, other IC foundries' VF implementations focus on providing customers with batch-based and summary information, so as to reserve manufacturing capacity from a desired IC factory in advance.

In summary, most of current VFs provide customers with the following functions:

1. Reserve capacity in advance [34].
2. Provide order-fulfillment related data such as on-line WIP, engineering and financial [9,16] reports.
3. Simulate experiments for specific critical operations to save cost [11].
4. Test scheduling simulations [17].
5. Test supply chain integration and simulation [31].

Table 1: Summary of prior related approaches

Sensor	Project	Accuracy (for % cases)	Advantage	Drawback
Access Point	RADAR [3]	400cm (~50%)	Easy Deployment	Delicate Calibration
RFID	LANDMARC [19]	200cm (~100%)	Enables 3D Localization	Imprecise Location Estimate and Signal Variations
Ultrasonic	Active Bat [14]	3-10cm (~95%)	High Accuracy	High Cost and Needs Time Synchronization
Infrared	Active Badge [37]	500cm (~100%)	Low Cost	Imprecise Location Estimate
Camera	EasyLiving [18]	30cm	No Need for Wearable Devices	High Cost and Delicate Calibration
Load Sensor	GRF [24]	N/A	No Need for Wearable Devices	Costly Deployment

2.2 Indoor Tracking Survey

Table 1 summarizes relevant prior work on indoor tracking along with some of their corresponding projects. For example, the LANDMARC system [19] spends approximately one minute per iteration to scan eight discrete power levels of the RFID tags in order to estimate the signal strengths of the active tags. After that, it uses a k-nearest-neighbor algorithm to determine the estimated coordinates of the unknown tags. The entire system operates under a basic assumption that all active tags have roughly the same RF signal emission strength. Thus, large variations in the signal strengths will cause erroneous results. The GRF (Ground Reaction Force) system [24] comprises several sophisticated load sensors with high resolutions. Rather than focusing solely on target tracking, its application concentrates more on movement recognition (such as jump, sit, stand, etc.).

Owing to the advances of RFID technology and the trend of utilizing RFID technology in supply chain applications among corporations, some companies (like [22]) have proposed the use of RFID tags for effective process manage in semiconductor fabrication to realize maximized productivity, reduction of data reading errors, speed-up of whole processes, steady decrease in labor forces, etc. On the other hand, some factory automation related corporations (like [23]) also proposed highly-reliable tracking solutions for supply chain applications to identify inventory in real-time and supply information for immediate decision-making; therefore, rather than leveraging other tracking techniques listed in Table 1, our proposed approach utilizes both RFID tags as a more practical way to track multiple targets inside a fab and it is also for addition of future enhancements.

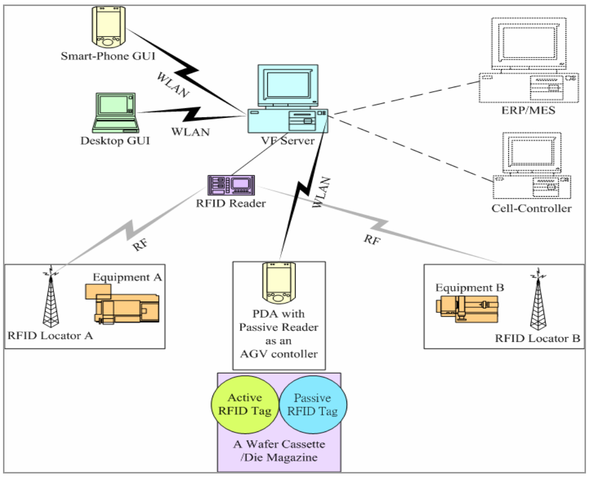

Figure 1: System overview

3. SYSTEM ARCHITECTURE

Figure 1 illustrates the overall system layout of our proposed VF. To simplify discussion, we show only two workstations, say station A and B. Additionally, each wafer cassette [7] or die magazine (a container of unbonded dies for die-bonding process in an IC assembly factory) has one active and one passive RFID tag [16] in which the same tag information is written (has a unique ID as its lot ID). Unlike a bar-code which is presently still commonly used in IC foundries, the content of an RFID tag can be read and updated wirelessly and dynamically.

Assume that a wafer lot is to be processed on station A first, and then transferred to station B by an AGV (Automated Guided Vehicle [13]) inside the same shop floor sub-area, such as an intra-bay [7]. The active RFID locators [16] with predefined sensing range are deployed on both stations and will cause any active RFID tags within the sensing range to send their tag information, including lot ID and locator ID, to the RFID reader which is connected to the VF Server.

Integrated with currently existing CC, MES (Manufacturing Execution System [7]) and ERP (Enterprise Resource Planning) systems, the VF server is responsible for gathering the information and then summarizing the output data for customer access. When the wafer lot moves out of the sensing range of locator A and B, the passive RFID reader on the AGV takes over responsibility for reading tag information and transmitting it to the VF server via a wireless protocol such as Wi-Fi.

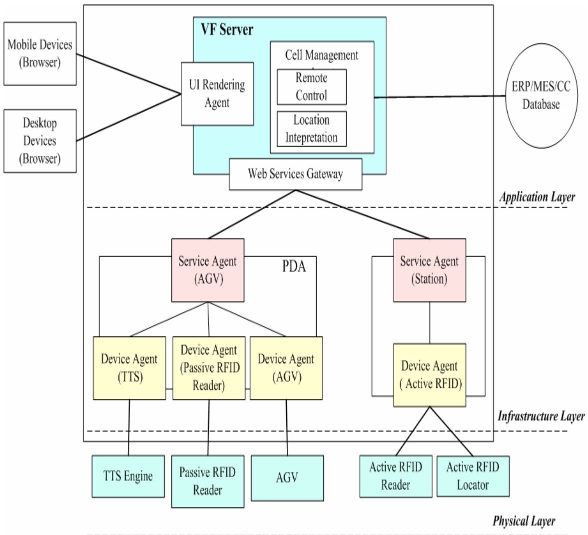

Figure 2: Software architecture

The software architecture of our VF system includes several components (see Figure 2). This 3-layered architecture is composed of four agent types with different responsibilities. Note that the database in Fig. 2 is a data repository integrated with currently existing ERP/MES/CC servers, thus making it feasible to provide real-time information to customers. A UI Rendering Agent provides information tailored to the capabilities of client's device types. For example, if one is using a handheld-device rather than a PC to browse the information, a UI Rendering Agent will send compact-sized web pages to the client.

A Cell Management Agent interprets location information and, additionally, may accept commands from clients for remote control; furthermore, it may access ERP/MES/CC related data if necessary to provide real-time or summarized production information. One or more devices on the Physical Layer are managed by a Device Agent in the Infrastructure layer.

Figure 3 illustrates the UML (Unified Modeling Language [9]) class diagram which shows the one-to-one or one-to-many relationships between the various agents. A Service Agent dynamically aggregates several Device Agents according to the specific needs of the Application Layer. Note that cooperation between Device Agents and Service Agents is bound at runtime, so that agents at the Infrastructure Layer are loosely coupled and are reconfigurable. The four agents in the left-bottom corner of Figure 2, which include three Device Agents and one Service Agent, are all installed on a PDA (Personal Digital Assistant). These Device Agents are in charge of activating TTS (Text To Speech [30]), reading passive RFID tags and controlling the simulated-AGV correspondingly. Each module in the software architecture is constructed as an autonomous agent and we utilize Web Services mechanisms to handle communication among agents.

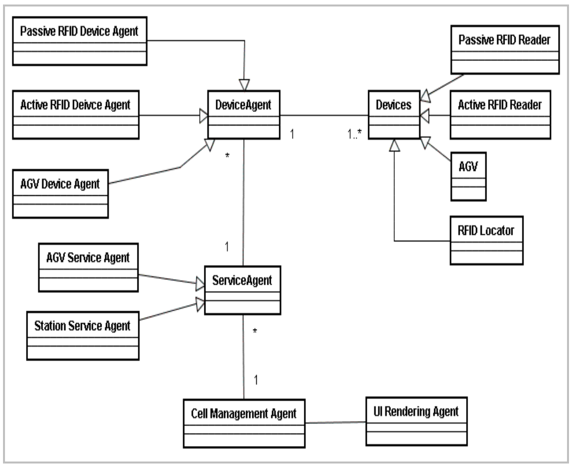

Figure 3: Agent UML class diagram

4. REAL-TIME FINE-GRAINED MULTI-TARGET TRACKING

4.1 Area Division

Most current VF systems provide, at best, station-based summarized information to their customers on a batch basis only several times a day. To provide real-time fine-grained lot-tracking inside a specific area such as an intra-bay, we use active RFID technology to divide an area into several sub-areas and gain access to a CC in order to gather more detailed information in that specific sub-area. Dividing an area into smaller sub-areas can assist with fine-grained lot tracking, especially for urgent-lot (or so-called "hot" lot) tracking, which is very essential for many important customers. For example, the wire-bonding area in an IC assembly factory has to be large enough to accommodate more than 80 or so bonding machines and many shelves for standby die magazines containing unbonded dies. This can make it difficult to rapidly locate a specific lot or magazine, unless we waste man-power of an operator to keep an eye on these lots, which is definitely not cost-effective.

By attaching an active RFID on a wafer cassette or die magazine, we can track that lot effortlessly inside a certain area. By adjusting the predefined sensing range, RFID locators can divide a production area into several sub-areas. For power-saving purposes, an RFID locator must transmit wake-up signals periodically or discretely to an active tag to make it send out its tag information, including its lot ID and the locator ID, to a nearby RFID reader. The shorter the transmission period, the more power these active RFID tags will consume. If an active tag stays within an overlapped area, it will send all IDs of locators whose sensing range covers the overlapped areas in an arbitrary order. We can easily know which locator sensing range the RFID-attached lot is in as long as the active RFID tag information can be correctly read by the RFID reader. Assuming that the sensing range of a locator is circular-like, the divided sub-areas will have three possible categories: independent, tightly coupled and hybrid, as shown in Figure 4.

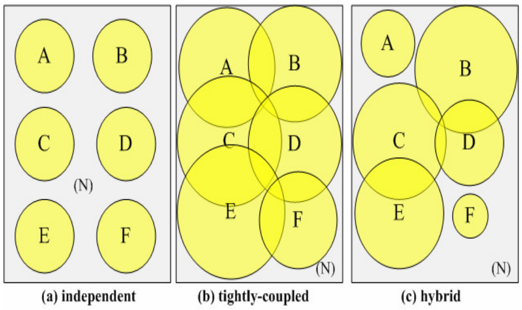

Figure 4: Three area division categories

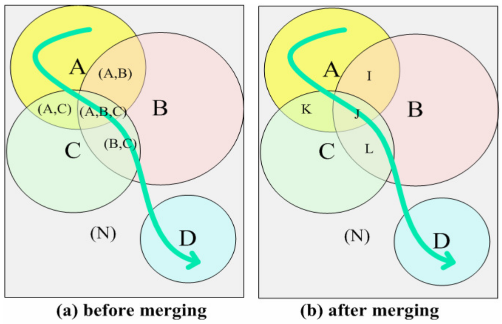

Figure 5: A wafer lot moves across sub-areas

Independent area-division is simple and heuristic but we will lose track of the lots located in the area (N), a no-signal area. Tightly coupled and hybrid can cover as much space as necessary, but they have to overcome the interpretation problem in overlapped areas. To simplify the problem, we assume that each locator has the same transmitting frequency to the active tags. For example, consider the case depicted in Figure 5 (a): A single lot, say lot 123, is carried from sub-area A, then crosses several overlapped sub-areas including (A,C), (A,B,C), (B,C), B, (N) to the destination D. In this case, we may receive one of the possible movement strings in Figure 7 (a). In the next section, we will present an approach to distinguish these overlapped sub-areas and make them separate from one another, as is shown in Figure 5 (b)

4.2 Sub-area Definition and Location Interpretation

We can define a sub-area Z_i as:

$$Z_i = (\sigma, \Gamma, \eta) \quad (1)$$

Where σ is a sub-area string set whose entities denote possible combinations of locator IDs perceived by a reader within a signal sensing interval T. Γ is the interval used to merge consecutive strings.

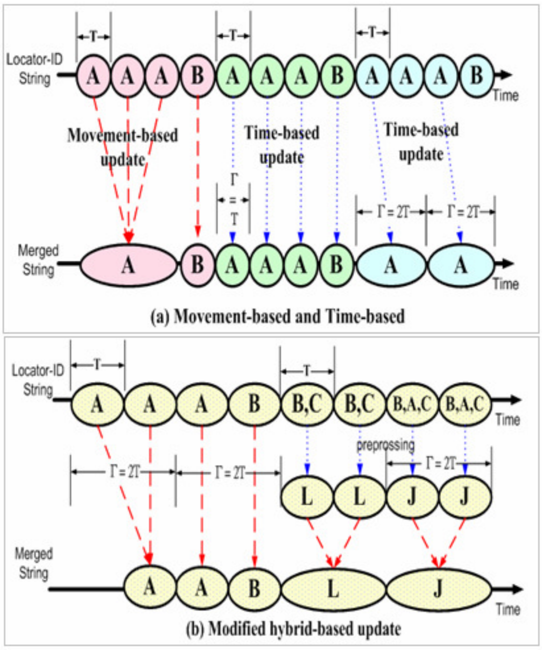

Figure 6: Three movement string merging schemes

η is the possible noise string in the sub-area based on noise model.

Take sub-area J in Figure 5 (b) as an example, its definition will be:

$$Z_J = (\{\text{“ABC”}, \text{“ACB”}, \text{“BCA”}, \text{“BAC”}, \text{“CAB”}, \text{“CBA”}\}, 2T, \eta_J) \quad (2)$$

We omit the effect of noise η here and will establish a model of it in the future work for future improvement. To simplify the model, the interval between two consecutive locator wake-up signals is the same, and equals T. σ is a set of the possible combinations of locator-ID strings (or movement strings) for a sub-area, and each entity of σ can be derived from three commonly used dynamic-path update schemes: movement-based [1], time-based [25] and hybrid-based [2]. The movement-based scheme updates a resultant movement string whenever a different locator ID is collected. On the other hand, the time-based scheme updates a resultant movement string periodically based on a preset interval. Although these two schemes seem reasonable enough and can be readily applied in our system, there are some inherent drawbacks that make them unsuitable for direct application here. For example, the time-based scheme cannot keep track of a lot which moves forward/backward too fast between two nearby sub-areas under a long update interval, whereas the movement-based scheme may overwhelm the whole system under the same conditions and will also lose the information about the duration time over which the lot stays in a sub-area.

Instead of applying these two schemes directly, our system adopts an existing hybrid-based update scheme which we slightly modify to meet our needs. Figure 6 illustrates how to condense the original movement strings based on these different schemes; therefore, the merged string can keep as many necessary characteristics as possible and also relieve the VF server from being overwhelmed. This filtering and merging task is performed by the Active RFID Device Agent in Figure 7. Assume that we have received the first string in Figure 7 (a) as shown below:

“AAAAAAAAACACACCBACBACBCBCBCBBN NNNNNNDDDDD”

Figure 7 (b) shows the results after individually applying these three schemes to parse the above string, which is then re-arranged in a shorter format. As you can see in the table, modified hybrid-based update shows more accurate and concise results. The increment in the number of overlapped sub-areas will make the merging process more intricate and time-consuming, thus greatly increasing the burden of the Active RFID Device Agent.

Lot ID: 123	(A)	(A,C)	(A,B,C)	(B,C)	(B)	(N)	(D)
1	AAAAAAAAA	CACAC	CBACBAC	BCBCBC	BB	NNNNNNN	DDDDD
2	AAAAAAAA	ACAC	ABCABCA	CBCBCBC	BB	NNNNNN	DDDDD
3	AAAAAAAAA	CACA	BCABCA	BCBBCB	BBB	NNNNNNN	DDDD
…	…	…	…	…	…	…	…

→ Time

(a)

Lot ID: 123	(A)	(A,C)	(A,B,C)	(B,C)	(B)	(N)	(D)
Time-based(Γ=2T)	AAAA	CC	CC	BB	B	NNNN	DDD
Movement-based	A	CACAC	BACBAC	BCBCBC	B	N	D
Modified hybrid(Γ=2T)	AAA	I	J	L	B	NNN	DD

→ Time

(b)

Figure 7: Movement string condensation

Since presently available RFID locators can only be adjusted manually, we cannot adjust the sensing range automatically via an agent. If, in the future, such an automatic adjustment mechanism becomes available, we will be able to adjust the sensing range of a locator to avoid inappropriate coverage or excessive overlap from distinct locators. In light of recent developments in object tracking, we could also go further and, additionally, predict the possible tracks of moving targets attached with active RFID tags within the sensing range of RFID locators in a fab. For these reasons, we use the LeZi-Update [4,41] approach to derive dynamic updated tries [32] and predict possible next movements for any interested objects inside the fab.

4.3 Active LeZi (ALZ) for Lot Location Prediction

Rather than delving into a lengthy background discussion on ALZ [12], a variation of LeZi-Update that we employ here, we demonstrate its core ideas directly using an example. Consider a movement string σ = {aaababbbbbaabccddcbaaaa}. It has 23 symbols in length after being processed by a hybrid-based update scheme. Table 2 enumerates its contexts (sub-strings) with symbol frequencies for Markov-model [26] orders 0, 1, 2.

Table 2: Contexts of order 0, 1, 2 with their frequencies

Type	Order-0	Order-1	Order-2
$\Omega \mid \omega(f)$	a(5) b(4) c(1) d(2)	a|a(2) a|b(2) b|a(1) b|b(2) d|c(1)	a|aa(1) a|bc(1) b|ba(1)

An entry of "Ω | ω(f)" means that the symbol Ω∈ σ* appears with frequency f in the context ω ∈σ*, where σ* is the regular grammar notation (or Kleen's Closure) for a sequence of zero or more symbols from set σ. In other words, f is the number of matches of the "ω.Ω" pattern in the history, where the dot represents string concatenation. ALZ parses σ based on its encoding algorithm and maintains a dictionary of such contexts in a more compact form by a trie as shown in Figure 8. Every node represents a context (here it means a locator ID), and stores its last symbol along with the relative frequency of its appearance at the context of the parent node. Each edge shows each node's prefix. The root, at level 0, represents a null context and a node can have at most |σ| children. Level n stores the necessary statistics for an order-(n−1) Markov model.

Based on the trie just constructed, we can utilize ALZ to determine the probability of any movement string or path of interest. For example, if we want to know the probability of receiving a locator ID "a" given the perceived symbols "aa"; that is, the probability of a wafer lot staying in the same sub-area "a" after it has been there for two consecutive intervals. By utilizing ALZ, we can calculate the probability of receiving an upcoming locator ID "a" by the following equation:

$$L(a) = p_2^a + \{(1-p_2^a)[p_1^a + (1-p_1^a)(p_0^a)]\} \quad (3)$$

where L(a) denotes the probability of receiving a locator ID "a" given a certain perceived locator string with length equal to 2 and P_n^a is the result of an order-n Markov model for the upcoming "a" by look up in Table 2 or the trie in Figure 8. First, we calculate the probability of "aaa" which represents the context of an order-2 Markov model. By tracing the trie backwards from leaf "a" in the left-bottom corner, one "a" tags along with two "aa" as its prefix; therefore, the probability P_2^a is 1/2.

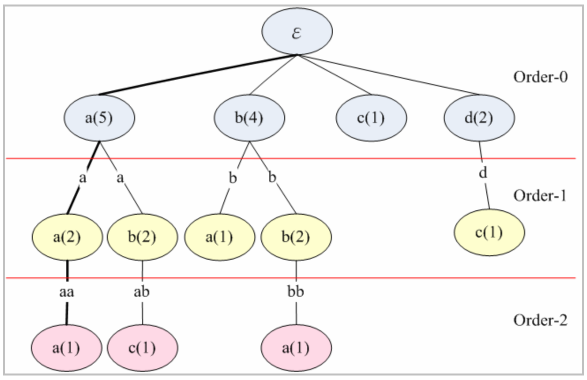

Figure 8: Trie-based movement string storage

Likewise, the probability of an order-1 Markov Model P_1^a is 2/5 (two "a" symbols follows five "a"s as their prefixes). Finally, the probability of an order-0 Markov model P_0^a is 10/23 because ten "a"s appear in the original σ string with a total of 23 symbols. ALZ can now calculate L(a) by substituting the variables in equation (3) for above results so that L(a) equals 1/2+1/2[2/5+3/5(10/23)] = 0.83.

After ALZ determines all the probabilities of possible paths, it then chooses a symbol (or locator ID) with the maximum probability from all possible paths. In our case, we can utilize the probabilities of these candidate movement strings to predict the next most likely sub-area in which a wafer lot or an operator may reside. This task is performed by the Station Service Agent and Location Interpretation Agent in Figure 2. Since every active RFID tag sends its tag data along with the locator ID to the reader, this makes multi-lot tracking feasible; however, we recommended applying this real-time multi-lot tracking mechanism only on urgent lots to avoid overwhelming the whole system with RFID tag information.

With track prediction, we can provide necessary production-related service or prohibit unauthorized behavior in advance. For example, a standby surveillance camera near an entrance can be switched on beforehand if some unauthorized visitors are approaching or a moving AGV may stop before colliding with a careless operator. In addition to benefits from track prediction, there are two extra advantages from area-division in an IC fab. First, we can roughly group similar workstations together within the same sensing range of one locator to avoid incorrect operations for urgent lots. If an urgent lot enters restricted sub-areas, the VF system will activate an alarm to warn the operator or Production Control personnel. Second, by densely deploying locators to cover the space of confidentially restricted regions, these active tags and locators can also be used to keep visiting guests or unauthorized operators at bay.

4.4 Extensible RFID-assisted Tracking Architecture

In order to mitigate the power consumption problem inherent from RFID tags and increase the accuracy of the whole tracking system, here we propose an extensible architecture (as shown in Figure 9) by means of the advantages from the use of multi-data fusion.

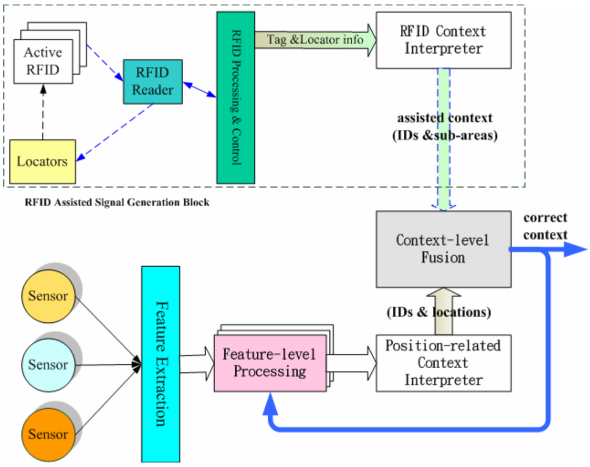

Figure 9: An extensible RFID-assisted tracking architecture

Fusion of data from multiple sensors has been proved effective in improving the accuracy of applications ranging from multiple-target tracking to very complex applications for multiple targets [36]. In addition to increased accuracy, combining observations from multiple sensors leads to improved estimates of the target status; moreover, the sensors in our proposed architecture can benefit from cooperation with one another and accomplish a common goal that possibly cannot be achieved simply by a single sensor source. Figure 9 depicts integration of the RFID system and other more accurate localization sensors (such as ultrasonic, RF, etc.). The RFID reader connects to the RFID Processing & Control (RPC) component whose main task is to retrieve the information from the RFID reader and control the frequency for acquiring the active tags based on hybrid-based update scheme introduced in sub-section 4.2.

This extensible architecture can integrate other location-related contexts from different sensors to improve the accuracy or even increase the features that can be discovered from the targets. Based on the feedback or the output context (which contains cassettes' IDs, their locations, and the reliability factors [20] reckoned by different sensors) from the Context Level Fusion (CLF) block, the RFID Context Interpreter (RCI) can determine if the system should wake up active tags by comparing the returned reliability factors with their corresponding predefined thresholds. If any fed-back reliability factor is lower than its acceptable thresholds, the RCI will start to trigger the active tags constantly via the locators. Regarding the function of the remaining blocks in Figure 9, please refer to [8,36] for more details.

Fusing data at a higher level, the CLF block is exempted from the responsibility of handling the detailed implementations of a variety of underlying sensors. Since most of the lower-level sensor data, such as video and load sensory data, are non-commensurate, they cannot be fused directly. In our proposed architecture, each sensor must individually provide a decision, which is subsequently combined to produce contextual information (including IDs, sub-areas or locations) for the CLF block to generate a feedback signal. In the current phase, the RFID system does not incorporate with other sensors; however, the proposed architecture is extensible and enables the synergistic use of the sensory data to achieve significant improvements or higher fault tolerances in the future. Additionally, another advantage of this architecture is that newly-created advanced techniques can completely replace the current RFID system as long as they can generate the same context (IDs and sub-areas) as the RFID system does.

As for detailed implementation of the CLF block for the location data, there are several practical techniques. From our survey, one possibility could be the use of a majority rule to select a candidate location with the maximum weighted sum. Another alternative is Bayesian-based fusion [20] where the reliability factor is defined for each sensor as a node in the Bayesian belief network to help the inference of a limited set of final possible locations. Another popular fusion method is the particle filtering-based approach [9], where the reliability factors can serve as parameters to adjust the likelihood of each particle representing the possible location for a target.

4.5 Implementation in Embedded-Devices

In order to implement a practical agent which accepts sensor inputs and provides outputs to actuators such that it can interact with its environment, we use a .NET Compact Framework [5,38] based HP iPaq [15] PDA (a powerful embedded hand-held device) to control the simulated-AGV with an 8-bit 8051 chip as its central controller and a serial communication wire as its control input. As shown in Figure 10, the PDA is equipped with a passive RFID reader as sensor input and an RS232 wire as actuator output. Figure 11 illustrates the system architecture on the PDA, which comprises several Device Agents integrated with a Service Agent. Note that we utilize a wrapper module to separate function modules from driver DLL (Dynamic Link Library) files, some of which are provided by vendors, so that revisions in driver DLL files will have the least impact on our function modules.

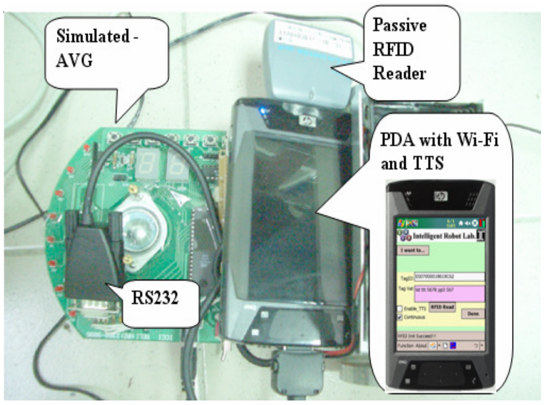

Figure 10: Simulated-AGV controlled by a PDA

The challenges and solutions of software development and integration on a .NET-based iPaq platform include:

1. The lack of many convenient built-in functions in the full-fledged .NET framework used on PC platforms: Since the .NET Compact Framework is like a subset of the full .NET Framework, we have to establish unsupported functions from scratch.
2. More time-consuming debugging procedure: A powerful step-by-step debugging and break–point debugging are only available on a Pocket Emulator instead of on the real target PDA device.
3. Latent bugs in the .NET Compact Framework: After iterating run-and-debug many times, we found that some latent bugs can only be resolved by patching the latest service-pack files from Microsoft's website.

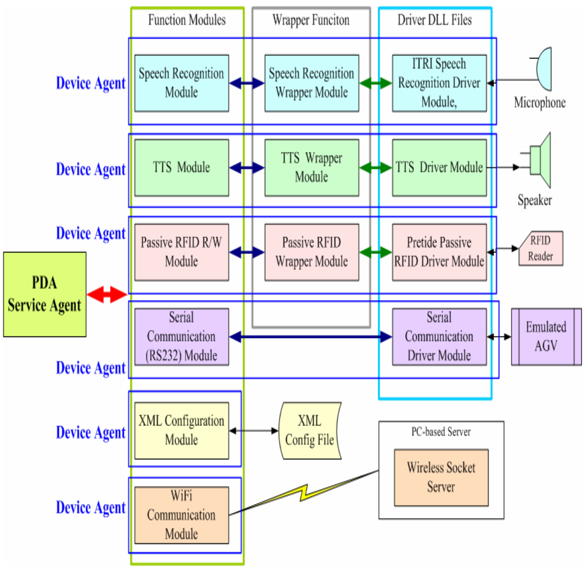

Figure 11: System architecture on the PDA

5. EXPERIMENTAL SCENARIO

We created an experimental scenario in order to illustrate the core ideas of our mobile VF system. In the hypothetical scenario, John, a manager from the Production Control department in TSMD Corp. is waiting for a flight to take off when he gets informed that he must pull in the delivery date by two days on two lots from an extremely important customer, TJ Corp. Under such short notice from an important customer, John performs the following steps in order to fulfill the order on schedule:

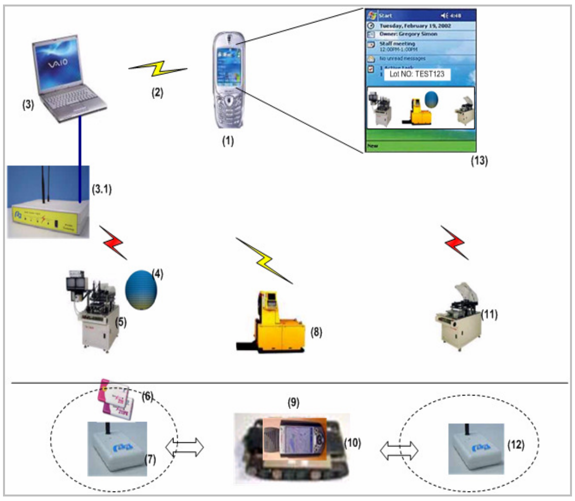

Figure 12: A demo scenario of our VF system

1. John uses his smartphone (Figure 12, (1)) to connect via Wi-Fi (Figure 12, (2)) to TSMD's VF system (Figure 12, (3)).
2. Realizing that two lots, say TJ123 and TJ456, should be processed as soon as possible to avoid severe complaints from TJ, John marks these two normal-priority lots as urgent lots. Since each cassette for important customers has an active and a passive RFID tag (see Figure 12, (6)) and can be traced by the VF system, the smartphone shows that TJ123 (Figure 12, (4)) is now waiting in a long line for machine A (Figure 12, (5)), which is inside the sensing range of a locator (Figure 12, (7)).
3. John looks up a summarized database table provided by querying the scheduling system in the MES, the VF shows another machine B (Figure 12, (11)) could be preempted by this urgent lot. He manually updates this urgent lot via his smartphone at once.
4. The VF server sends a remote command to update their MES, and CC which are assumed to be in the same server as the VF. The TTS engine equipped with AGV controller (Figure 12, (9)) on the AGV (Figure 12, (8) or (10)) will be activated to notify those operators nearby the AGV that some schedules have been revised. The AGV

receives a series of commands of SECS/HSMS [27], standard protocols for semiconductor equipment communication, and then commences to move TJ123 out from machine A, crosses several intermediate sub-areas and finally gets to the destination, machine B. The passive RFID reader attached on the AGV controller is used to read the passive tag when the AGV moves beyond the sensing range of RFID locators. After TJ123 has been re-arranged, John goes through the same procedures for lot TJ456.

5. All active RFID tag signals will be collected by an active reader (Figure 12, (3.1)), which is connected to the VF server.
6. The progress on these two urgent lots would be displayed simultaneously on the screen (Figure 12, (13)) of John's smartphone so that he can verify the expected behavior of his re-prioritization of these two lots.

After arranging these two urgent lots, John feels relieved to continue his trip.

6. CONCLUSIONS AND FUTURE WORK

6.1 Conclusions

In this paper, we have discussed and demonstrated an agent-based virtual fab architecture that provides a flexible and mobile method to monitor real-time multi-lot manufacturing status by using practical and readily available techniques such as RFID-enhanced embedded devices and mobile-accessible sever applications. The presented VF architecture has been simulated in our lab with a hypothetical scenario in order to illustrate the concepts and to emphasize the importance and feasibility of our proposed system.

In summary, our design has the following features:

1. Agent-based RFID applications: The applications are responsible for dynamically merging and interpreting incoming RFID reader messages as well as the output messages which report the current status of the production flow. Autonomous agents communicate and cooperate with one another to perform tasks on the VF system.
2. Real-time fine-grained multi-target tracking: RFID-enhanced technology makes it feasible to track multiple-targets inside a certain sub-area of the fab. This mechanism can also be used to improve the security level of a foundry.
3. Flexible user-interface: The web-based user interface provides the flexibility of monitoring production flow on both mobile and desktop devices.

6.2 Future Work

There are, however, several issues from our design which still need to be addressed in further studies:

1. RFID signal distortion from environmental interference: Noise is inevitable and wireless devices near the RFID readers or locators may interfere with a reader's precision when receiving correct tag information.
2. The cost of using RFID technologies: Presently available active RFID tags are still too expensive to be attached to every lot inside a foundry.
3. Power consumption from Active RFID tags: Power consumption has long been a major issue for handheld embedded systems. Each active tag is a battery-powered embedded system and often suffers power-shortage problems.

To overcome the above issues, we plan to conduct further studies to model noise and to construct more optimal policies for power-saving. To make the agents more extensible and scalable, we will utilize OSGi (Open Services Gateway Initiative [21]) as our future underlying infrastructure, since OSGi provides a Java-based lightweight container for dynamic software components which can interact with one another. The OSGi framework was originally designed to fit the needs of pervasive or mobile devices, which need run different software and be managed remotely. As a result, all of our agents can be hosted on that platform and communicate through Web Services mechanisms. Additionally, we will incorporate more sensor inputs with effective fusion methods for more accurate service-provision.

REFERENCES

1. Akyildiz, I. F. and Ho, J. S. M., 1995, "Movement-based location update and selective paging for PCS networks," *IEEE Transaction on Networking*, pp. 629-638.
2. Amiya, B. and Sajal, K. D., 2002, "LeZi-update: An information-theoretic framework for personal mobility tracking in PCS networks," *Wireless Networks*, Vol. 8, pp. 121-135.
3. Bahl, P. and Padmanabhan, V. N., 2000, "RADAR: An in-building RF-based user location and tracking system," *19th Annual Joint Conference of the IEEE Computer and Communications Societies.*
4. Bhattacharya, A. and Das, S. K., 2002, "LeZi-update: An information-theoretic framework for personal mobility tracking in PCS networks," *Wireless Networks*, Vol. 8, pp. 121-135.
5. Boling, D., 2003, "Programming Microsoft Windows CE.NET," *Microsoft Press.*

6. Chang, S. C., Chou, T. L., Guo, R. S., Su, Y. H. and Lu, L. L., 1998, "A dynamic binding model for service creation in virtual Fab.," *Proceedings of 1998 Semiconductor Manufacturing Technology Workshop*.
7. Chung, S. L. and Jeng, M. D., 2003, "An overview of semiconductor Fab automation systems," *IEEE International Conference on Robotics and Automation*, pp. 1050-1055.
8. Hall, D. L. and Llinas, J. ,2001, *Handbook of Multisensor Data Fusion*, CRC.
9. Fowler, M. and Scott, K., 2003, *UML Distilled: A Brief Guide to the Standard Object Nodeling Language*, Addison Wesley.
10. Fox, D., Hightower, J., Kauz, H. Liao, L. and Patterson, D. J., 2003, "Bayesian filtering for location estimation," *IEEE Pervasive Computing*, Vol. 2, pp. 24.
11. Garozzo, G. and La Magna, A., 2002, "The virtual Fab in microelectronics: The dry etch case," *The 6th World Multiconference on Systemics, Cybernetics and Informatics*.
12. Gopalratnam, K. and Cook, D. J., 2003, "Active LeZi: An incremental parsing algorithm for device usage prediction in the smart home," *Florida Artificial Intelligence Research Symposium*.
13. Groover, M. P., 2001, *Automation, Production Systems, and Computer-Integrated Manufacturing*, Prentice Hall.
14. Harter, A., Hopper, A., Steggles, P., Ward, A. and Webster, P., 1999, "The anatomy of a context-aware application," *The 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, pp. 1-15.
15. HP iPAQ Pocket PC hx4700 Series, http://h50025.www5.hp.com/hpcom/tw_tw/11_26_60_2750summary.html.
16. Finkenzeller, K., 2003, *RFID Handbook*, John Wiley and Sons.
17. Koriyama, H., Yazak, Y., Lhonbori, I, Kato, Y and Kisakibaru, T., 1995, "Virtual manufacturing system," *Proceedings of International Symposium on Semiconductor Manufacturing*.
18. Krumm, J., Harris, S., Meyers, B., Brumitt, B., Hale, M. and Shafer, S., 2000, "Multi-camera multi-person tracking for EasyLiving," *Third IEEE International Workshop on Visual Surveillance,* Dublin Ireland, pp. 1-8.
19. Lionel, M. N., Yunhao, L., Cho, L. Y. and Abhishek, P. P., 2004, "LANDMARC: Indoor location sensing using active RFID," *Proceedings of the 1st IEEE International Conference on Pervasive Computing and Communications*, pp 407-415.
20. Lo, D., Goubran, R. and Dansereau, R., 2005, "Robust joint audio-video localization in video conferencing using reliability information II: Bayesian network fusion," *IEEE Transactions on Instrumentation and Measurement*, Vol. 54, pp. 1541-1547.
21. Open Service Gateway Initiative, 2003, *Service Platform*, IOS Press.
22. RFID for Semiconductor Process Management, http://www.ceyon.co.kr.
23. RFID Tracking, http://www.brooks.com.
24. Robert, H. and Rupert, C., 2001, "Recognizing movements from the ground reaction force," *Proceedings of the 2001 Workshop on Perceptive User Interfaces*, ACM Press.
25. Rose, C., 1996, "Minimizing the average cost of paging and registration: A time-based method," *Wireless Networks*, Vol. 2, pp. 109-116.
26. Russell, S. J. and Norvig, P., 2003, *Artificial Intelligence: A Modern Approach*, Prentice Hall.
27. SECS/HSMS Standard, http://wps2a.semi.org/wps/portal.
28. Su, Y. H., Chang, S. C., Guo, R. S. and Lai, Y. C., 2000, "Application of dynamic manufacturing service provisioning mechanism to delivery commitment," *Proceedings of Semiconductor Manufacturing Technology Workshop*, pp. 106-117.
29. Su, Y. H., Guo, R. S., Chang, S. C., Chou, T. L. and Lai, Y. C., 1998, "Manufacturing service creation and management for next generation virtual Fab," *Proceedings of International Symposium on Semiconductor Manufacturing*.
30. The Industrial Technology Research Institute-TTS/SR Technology, http://www.ccl.itri.org.tw/English/E02-11.asp.
31. Torres, K., Smith, E., McDonald, C. and Cleavelin, C. R., 1999, "The virtual Fab the core of future technology development," *The 10th Annual IEEE/SEMI. Advanced Semiconductor Manufacturing Conference and Workshop.*
32. Trie Structure from Wikipedia, http://en.wikipedia.org/wiki/Trie.
33. Tseng, F. C., 1998, "TSMC's semiconductor strategy," *Proceedings of the Seventh International Symposium on Semiconductor Manufacturing,* pp. 5-7.
34. TSMC-eFoundry, http://alice.charlie.idv.tw.
35. Virtual Fab, http://www.cc.nctu.edu.tw/~smc/.
36. Waltz, E. L. and Llinas, J., 1990, *Multisensor Data Fusion*, Artech House Publishers.
37. Want, R. and Hopper, A., 1992, "Active badges and personal interactive computing objects," *IEEE Transactions on Consumer Electronics*, Vol. 38, No. 1, pp. 10.

38. Wigley, A. and Wheelwright, S., 2003, *Microsoft.NET Compact Framework*, Microsoft Press.
39. Yu, C. Y. and Huang, H. P., 2001, "Development of the order fulfillment process in the foundry Fab by applying distributed multi-agents on a generic message-passing platform," *IEEE/ASME Trans on Mechatronics*, Vol. 6, pp. 387-398.
40. Yu, C. Y. and Huang, H. P., 2001, "Development of virtual foundry Fab based on distributed multi-agents," *IEEE International Conference on Systems, Man and Cybernetics*, pp. 1030-1035.
41. Ziv, J. and Lempel, A., 1978, "Compression of individual sequences via variable-rate coding," *IEEE Transaction on Information Theory*, Vol. 24, No. 5, pp. 530-536.

ABOUT THE AUTHORS

Ching-Hu Lu received B.S. and M.S. degrees in Electrical Engineering from National Taiwan University Department of Science and Technology, Taiwan, R.O.C., in 1993 and 1995. He is currently a Ph.D. degree candidate in the Department of Computer Science & Information Engineering at National Taiwan University, Taiwan, R.O.C. He also has more than 6 years of experience working in the IC industry in Taiwan. His present research interests are in Home Automation, Factory Automation, Intelligent Space, and related topics.

Chun-Feng Liao received a B.S. degree in Computer Science from National Cheng-chi University in 1998,and an M.S. degree in Computer Science from National Cheng-chi University, Taiwan, R.O.C., in 2004. He is currently a PhD student in the Department of Computer Science & Information Engineering at National Taiwan University, Taiwan, R.O.C. His research interest is middleware for Context-Aware Smart Environments.

Chao-Lin Wu received a B.S. in Industrial Technology Education from National Taiwan Normal University, Taiwan, R.O.C., in 1996, and is currently a Ph.D. degree candidate in the Department of Computer Science & Information Engineering at National Taiwan University, Taiwan, R.O.C. His research interests are in Smart Home, Smart Environment, and Intelligent Space.

Li-Chen Fu received a B.S. degree from National Taiwan University in 1981, and M.S. and Ph.D. degrees from the University of California, Berkeley, in 1985 and 1987, respectively. Since 1987, he has been a faculty member and professor in both the Department of Electrical Engineering and Department of Computer Science & Information Engineering at National Taiwan University. He is now a senior member of both the Robotics and Automation Society and the Automatic Control Society of the IEEE, and became an IEEE Fellow (F) in 2004. His areas of research interest include robotics, FMS scheduling, shop floor control, home automation, visual detection and tracking, E-commerce, and control theory & applications.

(Received December 2005, revised November 2006, accepted December 2006)