Barcode Data Fusion through Application Level Events compliant middleware

Nikos Zarokostas
Sensap SA
Athens, Greece
nzaro@sensap.eu

Panagiotis D. Dimitropoulos
Sensap SA
Athens, Greece
pdimi@sensap.eu

John Soldatos
Athens Information Technology
Athens, Greece
jsol@ait.gr

George Stamoulis
University of Thessaly
Volos, Greece
georges@uth.gr

***Abstract*— In this paper we propose a mechanism for fusing data identified by a barcode scanner and sensor data through ALE compliant middleware. In this way the ALE server becomes an Auto-ID instead of RFID middleware, offering to capturing applications a unified view of the statements performed in a specific plant or warehouse, regardless of the technology used to capture these statements. The mechanism defines the notion of the generalized EPC, along with its corresponding generalized EPC pattern.**

EPC, ALE, TDS, Barcode Fusion

I. Introduction

The EPCglobal architecture framework [1] introduces a set of standards that define how RFID data is identified, captured and exchanged among the various entities that participate in the supply chain management. More specifically, the framework defines roles and interfaces that regulate the communication between tags and readers, readers and middleware, middleware and applications and applications with each other. A fundamental contribution of the EPCglobal architecture is the Filtering and Collection role, along with the interface it exposes to higher level applications, namely the Application Level Events interface [2]. The Filtering and Collection role is responsible for performing collection, filtering and aggregation of RFID information, generated by underlying readers, on behalf of applications that have declared their interest on the findings of those readers. External applications access the functionality of the Filtering and Collection role by utilizing the ALE API.

II. Problem Statement

When utilizing an ALE implementation for performing Business Activity Monitoring in the fields of Supply Chain Management and Warehouse Automation, a desirable feature would be to be able to integrate data identified by barcode scanners in the original flow of information from RFID devices towards the ALE server. In this way the ALE server becomes an Auto-ID instead of RFID middleware [4-6] offering to capturing applications a unified view of the statements performed in a specific plant or warehouse, regardless of the technology used to capture these statements.

For example, a typical pick and pack business process in the field of Supply Chain Management includes three individual statements: the identification of the location from which items are removed, the identification of a particular item and finally the identification of the container that this particular item is assembled in. If this process was executed using barcode technology, it would be time consuming since all picked items would be scanned one by one, in the case of picking different types of items from the same location, or just the first one with its corresponding quantity, in the case of picking the same item from a location. If on the other hand this process was executed using the RFID technology, it would be unreliable since multiple RFID tagged locations would be present simultaneously in the field of view of the antenna of the handheld device. In order to execute the process accurately using the RFID technology, spacing limitations would have to be imposed on the RFID tags attached to the locations, maintaining a minimum distance from one to another. Moreover, the transmit power of the RFID antenna would have to be tuned carefully and probably constantly since a low transmit power is desirable when scanning locations or densely populated containers, but a high transmit power is necessary when scanning the contents of a closed pack. The business process can be executed accurately and efficiently, if the location and container codes were identified via a barcode scanner, taking advantage of the inherent line of sight, and the RFID technology be utilized for the instant identification and counting of multiple items.

From a capturing application perspective that resides above the Filtering and Collection role of the EPCglobal architecture framework, the above mentioned business process would conclude into defining an event cycle specification with one logical reader, associated with either two physical devices, namely the RFID reader and the barcode scanner, or one physical device with two read points, namely the RFID antenna and barcode laser beam. The application would then subscribe to that event cycle and receive as a result event cycle reports, containing both RFID and barcode data. This approach is in contrast with the EPCglobal architecture framework, which states that data captured by barcode scanners should be processed by capturing applications separately, without flowing through the Filtering and Collection role, as depicted in the following figure.

978-0-7695-4813-5/12 $26.00 © 2012 IEEE
DOI 10.1109/RFID.2012.16

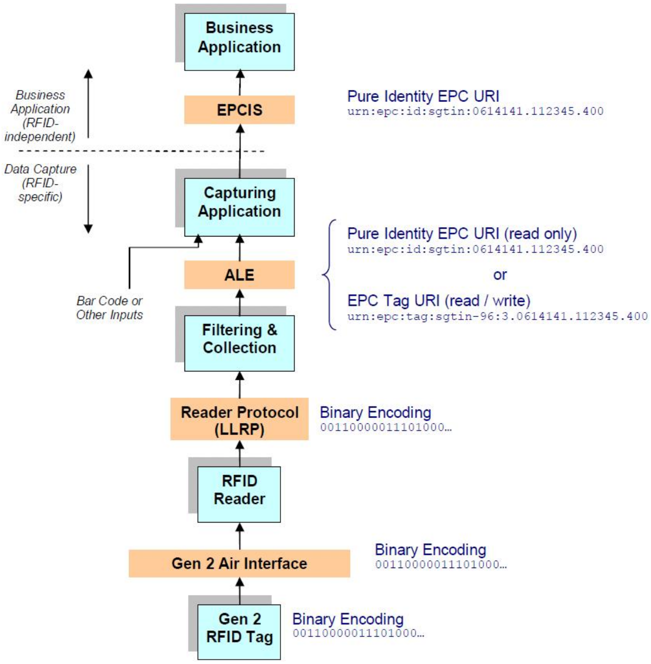

Figure 1. EPCglobal Architecture

III. Proposed Solution

In order to establish an event driven architecture that adopts the above mentioned requirements, we introduce the notion of the generalized EPC pure identity format. This format encompasses all GS1 application identifiers not defined in the EPCglobal Tag Data Standard specification [3], while at the same time complies fully with application identifiers included in it. The TDS specification defines among others, the representation of EPCs as URIs, in order for software applications to be able to manipulate EPC data in a way that is independent from low level tag encodings, which may vary, for example, in the number of encoding bits. The specification defines four categories of URIs. The first category of URIs represents pure identities, the second represents specific tag encodings, the third represents EPC patterns and the fourth represents raw tag information. The proposed generalized EPC URI can be considered as either a fifth category or an extension to the first and third one.

Moreover, we introduce an algorithm that enables the transformation of GS1-128 barcodes into generalized EPC pure identities, enabling an ALE implementation to perform the same filtering and grouping operations to the transformed barcode, exactly as it would apply them for an actual EPC. Finally, we introduce an algorithm that states when a generalized EPC matches with a generalized EPC pattern, just as an EPC matches with an EPC pattern.

These three concepts, allow the definition of event cycle specifications that contain generalized EPC patterns for filtering and grouping purposes and the generation of event cycle reports that contain generalized EPCs as members in their EPC group list, making it possible to achieve a seamless integration of barcode scanner data with RFID reader data, under the ALE interface abstraction layer, as depicted in the following figure.

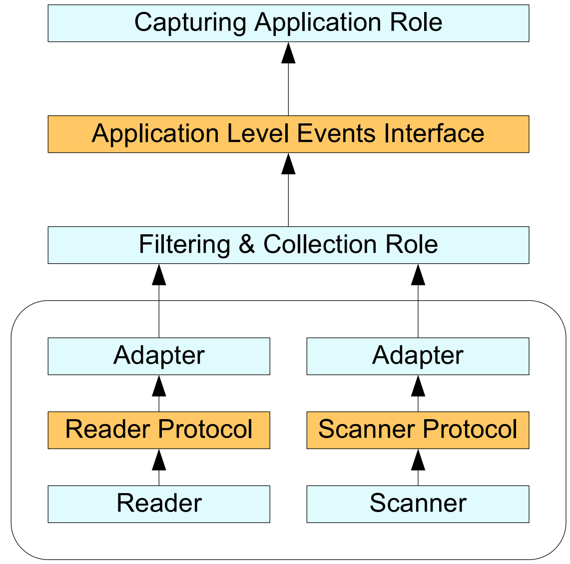

Figure 2. Proposed Architecture

IV. Architecture

A. *Generalized EPC Definition*

A generalized EPC pure identity has the following regular expression urn:epc:id:gs1:manager.[AI.value]+, where manager is the unique number that identifies the organization that owns the product, followed by one or more pairs of GS1 application identifiers along with their corresponding values. Examples of generalized EPCs are the following: urn:epc:id:gs1:52120005.01.1234 for the GTIN barcode 05212000512345, which identifies a trading item of an organization; urn:epc:id:gs1:52120005.01.1234.10.A1234 for the GS1-128 barcode (01)05212000512345(10)A1234, which identifies a trading item of an organization, along with its lot number; urn:epc:gs1:52120005.01.1234.10.A1234.3104.123456 for the GS1-128 barcode (01)05212000512345(10)A1234(3104)123456, which apart from the trading item and lot number reveals also the weight of the particular item class. From the above examples it is evident that the generalized EPC pure identity format extends the TDS specification, as it encompasses all combinations of GS1 application identifiers, regardless of whether those combinations can be encoded to an RFID tag or not.

A generalized EPC pattern has the following regular expression urn:epc:idpat:gs1:manager.[AI.value]+, where in this case the value can be an alphanumeric that matches the regular expression \w+, or a range that matches the regular expression \[\w-\w\] or the symbol *, just as in an normal EPC pattern. A generalized EPC pattern represents a class of generalized EPCs in the exact same way an EPC pattern represents a class of EPCs. Examples of generalized EPC patterns are urn:epc:idpat:gs1:52120005.01.*, which represents all trading items administered by the organization with the unique manager number 52120005;

urn:epc:idpat:gs1:5212005.01.1234.10.* which represents all trading items of type 1234 of the organization 52120005, independent of the lot number; urn:epc:idpat:gs1:52120005.01.1234.10.A1234.3102.[12345 0-123456] which represents all trading items of type 1234 with lot number A1234 and weight in the range of 1234,50 Kg to 1234,56 Kg.

From a software perspective a UML diagram that depicts the proposed extension in the hierarchy of URI representations of EPCs is as follows.

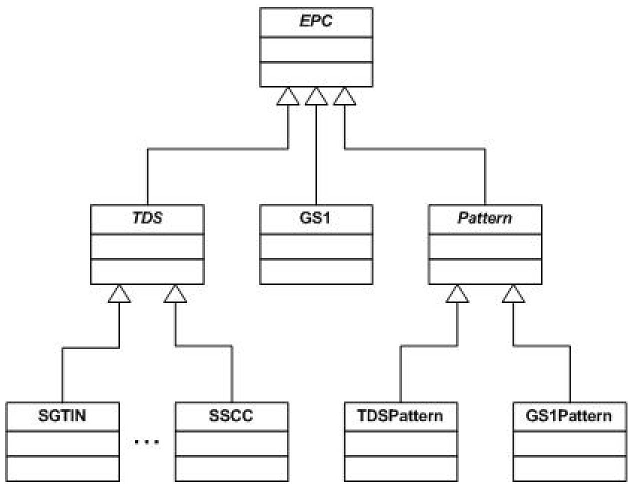

Figure 3. Generalized EPC UML diagram

B. Generalized EPC Matching

The ALE specification denotes that an EPC matches an EPC pattern if every field of the EPC matches the corresponding field of the EPC pattern. The EPC field value matches the pattern value if the pattern value is decimal and equal with the EPC value, or if the pattern value is a range and the EPC value is included in that range or the pattern value is a star. The above algorithm is extended to apply for generalized EPC and generalized EPC patterns in the following way: a generalized EPC matches a generalized EPC pattern if they both contain the same set of application identifiers and the value of every application identifier of the generalized EPC matches with the corresponding value of the generalized pattern. The matching of the two values is performed in the exact same way as denoted in the ALE specification, with the only difference being the presence of alphanumeric values. For example, the generalized EPC pure identity urn:epc:id:gs1:52120005.01.1.10.2 matches with the generalized EPC pattern pure identity urn:epc:idpat:gs1:52120005.01.*.10[1-3], while is does not match with the generalized EPC pattern urn:epc:idpat:gs1:52120005.01.2.10.*.

C. Generalized EPC Integration

Upon the reception of an EPC from an underlying physical reader, an ALE implementation must match that EPC against the EPC include and exclude filter patterns. If the EPC matches with one of the include filter patterns and does not match with any of the exclude filter patterns contained in the defined event cycle specification, then the given EPC is said to survive filtering and it will be reported in the generated event cycle report. The next step for the ALE engine is to match the given EPC against all group patterns contained in the event cycle specification and report it to the subscribed application inside that group.

In order to extend the ALE engine to fuse GS1-128 barcodes a role is needed that will undertake the task of transforming these GS1-128 barcodes to generalized EPCs. If this transformation is achieved then the above mentioned operations defined by the ALE specification apply also for the generalized EPCs, due to the introduction of the generalized EPC matching, and as a consequence for the corresponding GS1-128 barcodes. Therefore if an event cycle specification is defined, which contains both EPC patterns and generalized EPC patterns as include, exclude and group filters, then the corresponding event cycle reports would fuse RFID and barcode data in a seamless way.

In order to achieve this transformation a regular expression is produced from each generalized EPC pattern. This regular expression is generated in the following way: for every application identifier of the generalized EPC pattern the corresponding GS1 regular expression is retrieved, as defined in the GS1 general specification. The application identifier is appended as a prefix in the GS1 regular expression and if the application identifier does not have a predefined length, according to the GS1 specification, and is not the last then the group separator ASCII character is appended as suffix. The final composite regular expression is derived from the concatenation of the individual regular expressions produced in every iteration.

Afterwards, the GS1-128 barcode is matched against the generated regular expressions of all include, exclude and group filter generalized EPC patterns contained in the event cycle specification. If a GS1-128 barcode matches with the generated GS1 regular expression, then the composite GS1-128 barcode is parsed according to the GS1 general specification into its individual barcodes. The individual barcodes are processed together with the manager number derived from the EPC pattern from which the composite regular expression was originally derived. If the individual barcodes constitute a combination defined in the TDS specification, as for example a GTIN barcode along with a serial barcode, then the algorithm dictated in the TDS specification is applied and a normal EPC is generated. If the combination of barcodes is not dictated in the TDS specification then a generalized EPC is generated, by concatenating the manager number, along with the AI value pairs. From now on the ALE engine handles all EPCs in the same way and applies the principals and operations that the ALE protocol dictates.

For example the generalized EPC pattern pure identity URI urn:epc:idpat:gs1:52120005.01.5294.10.321.21.* would produce the composite GS1 regular expression (()(01)(\d{14}))(()(10)(\w{1,20}))((\W+)(21)(\w{1,20})) that would match against the GS1-128 barcode (in human readable format) (01)05212000552945(10)321(21)123456. Therefore, the composite barcode would be decomposed into its individual barcodes (01)05212000552945, (10)321 and

(21)123456 and since the combination of a GTIN, LOT and SERIAL application identifiers is not defined in the TDS specification, a generalized EPC would be generated as urn:epc:id:gs1:52120005.01.5294.10.321.21.123456.

V. IMPLEMENTATION

The above mentioned architecture was evaluated during the two pilots that were conducted in the scope of the RFID-ROI-SME ICT-PSP No. 250438 EU funded project. The first pilot involved a company that operates in the packaging industry, while the second a company that operates in the apparel industry. The goal of the two pilots was to demonstrate a successful adoption of the RFID technology for small medium enterprises (SMEs), by the seamless integration of RFID data into their existing business processes.

In both cases the notion of the generalized EPC offered a unified view of the preinstalled barcode equipment and the newly installed RFID equipment. The above mentioned architecture enabled the installation of an ALE server inside the premises of each company, that performed filtering and collection of both RFID and barcode data. The middleware was configured by defining logical readers, comprised of both barcode scanners and RFID readers, along with event cycle specifications that orchestrated their operation in the context of the various business processes each company would execute. The architecture enabled the solution providers to receive asynchronous notifications in the form of event cycle reports, regardless of the underlying infrastructure making it possible to focus on the actual business logic. Moreover, existing business processes that were originally executed by barcode technology could seamlessly incorporate RFID data, while newly designed business processes that took advantage of the existence of RFID tags, could still be executed by barcode scanners.

VI. CONCLUSION

The introduction of the generalized EPC enables solution providers to fuse data captured by barcode scanners through ALE compliant middleware. In this way applications are able to utilize the same abstraction layer for receiving data from both RFID readers and barcode scanners, achieving an even more efficient execution of complex business processes. Moreover, capturing applications lying on top of the Filtering and Collection Role obtain data captured by barcode scanners as pure identity URIs, offering a unified approach for retrieving information about items of the physical world, regardless of whether an RFID tag or a barcode label has been attached to those items. Finally, by fusing data captured by barcode scanners through ALE compliant middleware, principals and operations defined in the ALE specification can be applied to barcode data, resulting in the reuse of filtering and grouping capabilities for an even wider range of information.

Fusing barcode and RFID data through EPC ALE compliant middleware complies with the industry's widely adopted belief that the RFID technology is not a replacement for barcode technology, rather a very useful addition. Neither of the two technologies can replace one another, instead they can coexist and be utilized transparently according to the needs of each particular deployment. The low cost, inherent line of sight and wide penetration of the barcode technology are only some of the advantages that cannot easily be replaced by RFID. Therefore, the proposed solution enables the interchangeable use of the two technologies, by architecting middleware that accepts both inputs.

ACKNOWLEDGMENT

This work was supported by the RFID-ROI-SME ICT-PSP No. 250438 EU funded project.

REFERENCES

[1] Architecture Review Committee, "The EPCglobal Architecture Framework version 1.4," EPCglobal, December 2010.

[2] Architecture Review Committee, "The Application Level Events (ALE) Specification, version 1.1.1," EPCglobal, March 2009.

[3] Architecture Review Committee, "GS1 EPC Tag Data Standard, version 1.6," EPCglobal, September 2011.

[4] Christian Floerkemeier, Christof Roduner, and Matthias Lampe, "RFID Application Development with the Accada Middleware Platform," IEEE Systems Journal, vol. 1, pp. 82–94, December 2007.

[5] The ASPIRE FP7 Project, http://www.fp7-aspire.eu.

[6] Electronic Publication: Digital Object Identifiers (DOIs):

[7] Michel Cezon, Guillaume Vaudaux-Ruth, Luc Laurens, John Soldatos, et. al., "Review of State-of-the-Art Middleware," ASPIRE Project Public Deliverable D2.1, June 2008.