Typing within the GS1 System Finding

This finding from the GS1 Architecture Group addressed the questions arising from mass detection of a large number of similar objects which are identified with the same GS1 Key

Release 1.0., Final, Mar 2015

Document Summary

Document Item	Current Value
Document Name	Typing within the GS1 System Finding
Document Date	Mar 2015
Document Version	1.0
Document Issue	
Document Status	Final
Document Description	This Finding from the GS1 Architecture Group addressed the questions arising from mass detection of a large number of similar objects which are identified with the same GS1 Key

Contributors

The topic has been discussed and worked out within the whole GS1 Architecture Group. Therefore, it is refrained from naming individual members here. For the list of current GS1 Architecture Group Members, please see *http://www.gs1.org/docs/gsmp/GSMP_Governance_Group_Members.pdf*.

Log of Changes

Release	Date of Change	Changed By	Summary of Change
1.0	04 March 2015	Andreas Füßler	Summarising discussions and results on Request for Findings of concern in Architecture Group calls from 19 Nov 2014, 03 Dec 2014, 17 Dec 2014 and 14 Jan 2015.

Disclaimer

Table of Contents

1 Question of Concern

The request for finding of concern asks for an answer to the following question:

"How can a subset of relevant objects with specific same characteristics (e.g. small load carriers) be detected in real-time in case of automated mass detection of a large number of similar objects which are identified with the same GS1 Key (e.g. a pallet containing different small load carrier types of which each item is marked with a GRAI)?"

2 Context

Concrete examples of how this question arises in an operational process can be pictured by:

1. A pallet having many different "load carriers", each identified with a GRAI (including serial number) and scanned by an RFID reader. The system must determine, within some very short fixed interval of time, how many of those load carriers are of a specified "type".
2. Load carriers are passing along a conveyor. The system reads the GRAI and must determine the "type" of the load carrier within some very short fixed interval of time, and send a signal to a diverter vane on the conveyor to send the carrier down one lane or another depending on the "type" that is determined.

A “type” in this context might be for instance:

a. The "type" is a characteristic of the load carrier itself (e.g., is it a small, medium, or large container). In this case, the "type" is determined at the time the GRAI is assigned to the load carrier. Different GRAI values (not counting the serial number) could conceivably be used to differentiate the "type" as needed by the application.

b. The "type" is a characteristic of the “contents” of the load carrier (e.g., does it contain kittens or radioactive waste). In this case, the "type" is determined at the time the carrier is loaded and may change as the carrier is reused. The GRAI without the serial number cannot be used by itself to infer the "type", but the GRAI plus its serial number could conceivably be used to look up the "type" in some system that received information about the "type" at the moment the carrier was loaded.

3 Potential Solutions and Assessment

The GS1 System Architecture provides at least two ways to solve the problem:

3.1 Solutions 1a: By Giving Each Load Carrier a Unique GRAI + Serial Number

At the place where process (1) or (2) (see chapter 2 above) is carried out, the GRAI + serial number can determine the "type" using a backend system for a look up.

However, as it is pointed out in the request, the latency of this lookup is potentially a problem. There are a variety of ways to address this latency; basically they are all around the caching of the lookup information in the front-end system that carries out the operational process. For example, receipt of a despatch advice might trigger the backend system to push the necessary GRAI type information to the front-end system's cache in anticipation of receiving the load carriers.

Prerequisite is to derive a physical state from booking information pretty correctly in advance. In cases where the physical flow is in control of the process this approach has limitations.

3.2 Solutions 1b: By Giving Each Load Carrier a Unique GRAI + Serial Number

The "type" can be encoded directly in the data carrier using a "supplementary data" Application Identifier (AI). (In a bar code, this requires use of a symbology that can handle the additional AI; in

RFID this requires an RFID tag with User Memory.) The front-end system thereby reads the "type" directly from the data carrier and does not need to consult a back-end system.

Feasibility is inter alia dependent upon price and technical performance of the RFID tags.

3.3 Comparison of Solution 1a with 1b

The details of solution 1a and 1b vary slightly depending on whether "type" is a function of the load carrier or the content (called (A) and (B) in chapter 2 above). In the case of (A), the same GRAI (without serial) for each distinct "type" should be used, so that the "type" can be looked up as a function of the GRAI (without serial number). This makes solution 1a easier as the lookup table is probably small and infrequently changing. It also makes solution 1b practical for bar codes.

Solution 1b is not as practical for bar codes in the case of (B) as the supplementary AI value would have to change each time the container is reloaded.

In the case of (B), the choice of GRAI (without serial) is not so important because the lookup is based on GRAI + serial number.

3.1 Solutions 2: Use the RFID Filter Value to Encode the "Type", and then Proceed in a Manner similar to Solution 1b

This solution would require the definition of corresponding filter values for GRAIs.

The limitations of solution 2 are:

- It is an RFID-only solution; bar codes do not have filter values. However, the applications in the context of the request are RFID based only.
- The filter value only has seven possible values, and these values can only be assigned by revising the EPC Tag Data Standard. So it does not scale to handle "type" as posited in the Request for Findings, where the number of possible "types" could be very large, be application or user specific, and could change frequently over time.
- For an economical handling of filter value definitions it would be advisable to classify object “layers” instead of concrete object types in case further filter values would be defined in GSMP in future.
- The filter value is not intended to convey business information. The Tag Data Standard says this (in Section 10):

> *„The filter value is additional control information that may be included in the EPC memory bank of a Gen2 tag. The intended use of the filter value is to allow an RFID reader to select or deselect the tags corresponding to certain physical objects, to make it easier to read the desired tags in an environment where there may be other tags present in the environment. For example, if the goal is to read the single tag on a pallet, and it is expected that there may be hundreds or thousands of item-level tags present, the performance of the capturing application may be improved by using the Gen 2 air interface to select the pallet tag and deselect the item-level tags. ...*
>
> *It is essential to understand that the filter value is additional "control information" that is not part of the Electronic Product Code …*
>
> *Because the filter value is not part of the EPC, the filter value is not included when the EPC is represented as a pure identity URI, nor should the filter value be considered as part of the EPC by business applications. ... the purpose of the filter values is to assist in the data capture process, and in most cases the filter value will be of limited or no value to business applications. The filter value is not intended to provide a reliable packaging-level indicator for business applications to use.“*

(See *http://www.gs1.org/sites/default/files/docs/epc/TDS_1_9_Standard.pdf*, p. 60)

A Appendix - The Request for Finding

Request for Finding – Brief Summary (one phrase or sentence)
How can a subset of relevant objects with specific same characteristics (e.g. small load carriers) be detected in real-time in case of automated mass detection of a large number of similar objects which are identified with the same GS1 Key (e.g. a pallet containing different small load carrier types of which each item is marked with a GRAI)?

Submitter Name	**1) Jens Vialkowitsch** **2) Jörg Sandlöhken**
Submitter Company	**1) Robert Bosch GmbH** **2) REWE-Informations-Systeme GmbH**
GS1 Member Organization of submitter (if known)	**GS1 Germany**
Submitter e-Mail	**1) *Jens.Vialkowitsch@de.bosch.com*** **2) *Joerg.Sandloehken@rewe-group.com***
Submitter Telephone	**1) +49(711)811-22694** **2) +49(221)149-2125**

A.1 Statement of Question or Concern (please be specific as to what you want answered)

Due to the increasing automation of logistics and manufacturing processes through the application of cyber physical systems the data volume in backend systems is constantly rising. This effect is reinforced by the reason of proliferation of auto-ID technologies in general and of RFID technology in particular.

Exemplified by the inventory management of containers, being an important process in logistics. In the inventory management process it is often sufficient to know e.g. which container type has left the warehouse or which type remains in a given area of the warehouse.

In today's process data are being recorded in the front end and then used to send requests to the backend system. The result of the evaluation in the backend system is then retransferred to the front end.

In the interaction between front end and backend increased granularity in data collection induces considerable data traffic and thus a loss of time. Furthermore, large amounts of data have firstly to be collected even though this data is discarded after the evaluation. This procedure appears to be inefficient.

Consequently, the following question arises:

What solution does the GS1 portfolio contain to enable users to filter objects of the same sub-type directly at the front end (e. g. RFID reader)? What is the best practice at the front end?

Example: A pallet contains different types of small load carriers. All carriers are identified with a GRAI. How can I filter out a particular small load carrier type **at the front end**?

This issue applies not only to containers, but also to almost all business objects. In the context of GTIN this fact has been taken into account by the allocation of different filter values and concerning assets there were recently introduced filter values to distinguish different types. However, the filtering possibility is being provided only for RFID and due to the available 3-Bits only seven different filtering options can be used. In addition to the current discussion in the context of RFID, similar challenges in relation to optical data media are not inconceivable (e. g. automated data capture in tunnel scanning).

In the present context the focus is on container management, however, the general approach will not be restricted to container management. It is rather the objective to draw general recommendations how to handle this issue in light of Internet of things, self-monitoring of objects and progressive automation. Supplementary information: The VDA (Association of the German Automotive Industry) has recognized the issue and started a standardization initiative within ISO.

A.2 Relevant GS1 Standards or other GS1 System Components (omit if unsure)

(S)GTIN, GRAI, SSCC, TDS, GenSpecs

To be filled in by Architecture Group			
Request #	**Date Submitted**	**Date Accepted for Consideration**	**Date Completed**
14-001	**2014-11-13**	**2014-11-19**	**2015-03-04**
Link to Architecture Finding		**http://www.gs1.org/docs/architecture/2015_03_6_RFF_Typing_and_GS1.pdf**	