

A Fusion of Data Analysis and Non-Monotonic Reasoning to Restore Missed RFID Readings

Peter Darcy, Bela Stantic, Abdul Sattar

*Institute for Integrated and Intelligent Systems, Griffith University
Queensland, Australia*

{P.Darcy, B.Stantic, A.Sattar}@griffith.edu.au

Abstract—Radio Frequency Identification (RFID) is a wireless technology which can efficiently track various items within certain proximity. It has the potential to become a great asset across many applications such as a tracking inventory within a warehouse and the ability to track medical utensils within a hospital environment. Unfortunately, there are several problems that hinder the wide scale adoption of RFID technology including the serious threat of missed readings. Current state-of-the-art methodologies which attempt to solve the problem of false negatives can still not effectively restore the data set completely. In this paper, we propose an architecture that utilises a fusion of both intelligent data analysis of the observational records and a non-monotonic reasoning engine designed to determine the most likely values to restore. We then perform an analysis upon our methodology in which we discuss the adoption of our application.

I. INTRODUCTION

Radio Frequency Identification (RFID) is technology that combines the use of readers and tags to wirelessly identify items. All items that are desired to be tracked are tagged with a transmitter which will respond by sending its unique identifier back to the reader when interrogated. There is a wide range of potential applications that may benefit greatly from the use of RFID. These range from ensuring the safety of transporting goods within a supply chain to tracing the current position of medical utensils within a hospital, all of which benefit commercial sectors of the community by cutting the costs of manual labour and time. The passive RFID architecture, which is the most cost-effective and practical tag system to implement, suffers from several problems that hinder the world-wide adoption of this technology. One of the core problems associated with this technology is missed readings within the data warehouse.

To avoid this crucial problem, researches have implemented several methodologies which can be utilised to prevent or correct missing observational data. These include physical solutions which attempt to ensure the tags are read, middleware solutions that involve filtering the oncoming data and deferred solutions that usually involve the correction or transformation of the data after it has been stored. We believe that, although all of these methodologies provide a certain level of missing observation prevention, the data cannot remain at an adequate level of integrity for meaningful analytical

uses. Disadvantages such as anomalies persisting in physical solutions, middleware applications not taking into account past and future readings and deferred solutions lacking intelligence undermine the benefits each previous methodology puts forth.

In this paper we propose a methodology that fuses the use of data analysis techniques and non-monotonic reasoning. Both of these techniques have been tailored for restoring RFID missing data. We go on to discuss the benefits of our method for restoring missing RFID data. We assume that the equipment utilised in our application are passive tags which are read individually by sequential readers within a static environment.

The remainder of the paper is organised as follows: Section II will focus on a discussion of what Radio Frequency Identification and Non-Monotonic Reasoning is. In Section III we summarise work regarding the correction of observational data and the motivation for our work. Our proposed methodology will be detailed in Section IV which will be followed directly by our experimental evaluation and analysis in Section V. Finally, in Section VI we conclude this document and investigate potential paths for future work.

II. BACKGROUND

Radio Frequency Identification (RFID) is a technology that utilises radio technology to identify a collection of objects. There is a wide variety of options when selecting the hardware ranging from expensive active tags to cheap and practical passive tags. Unfortunately, there are several anomalies, which prevent the technology from being utilised commercially on a universal scale. One of the most important anomalies is the missed readings, which is particularly associated with the passive tags. To minimise these anomalies, an intelligent correction algorithm needs to be put in place so that if or when a missed observation occurs, the system can restore the data to a useable level. Non-Monotonic Reasoning is an intelligent deterministic logic which is utilised to find the ideal solution among all the other solutions available when given various input data.

A. RFID

Radio Frequency Identification technology consists of a transmitter, antenna and computer to automatically identify

various tagged items. This technology is already utilised in many systems, such as the Post Office [1], Aviation Industrial Tracking [2] and Health Care [3]. The means by which RFID applications may be operated relies on an individual to place transponder tags onto items they deem necessary to track and utilise a reader to scan the area of these tags, at which point the tag will send its identification number. After the reader has collected this data, it is then stored in the middleware to be transferred to a data warehouse.

There are three distinct types of tags to choose for use, which are active, semi-active or passive. Active and semi-active are tags that derive power from a battery source either completely as in the case of the former, or only to extend the range of scan as with the latter. Both of these tags have a distinct flaw however, as it is necessary to draw power from a finite source which will shorten the life expectancy and also increase the cost of production. The passive tag, in comparison, does not rely on any battery source but rather utilises the electro-magnetic pulse generated from the reader to send its identifier back to the interrogator [4]. This means that, theoretically, the passive tag will last indefinitely and will also be cost effective in that the price for a tag will be only 5.8 cents USD [5].

Despite the many advantages that passive RFID tags possess, there are also many problems which arise from the architecture. These problems come in the form of low level data [6], error-prone recordings, high volumetric intake [7] and complex spatial-temporal aspects [8]. With regards to the error-prone nature of the system, there are three core anomalies that prevail within the data set, namely: duplicate readings, wrong readings and missed readings. Duplicate readings refer to instances in which a tag is recorded twice where it should only have been once [9]. Wrong readings, also known as false positive readings, are scenarios in which a tag is read where it should not have been and does not completely reflect reality [10]. Missed readings, also known as false negatives, are situations in which a reading was supposed to have taken place but has not been recorded from various reasons such as tag collisions, tag detuning, water/metal interference and misalignment of tags. It has been estimated that in a usual RFID system, only 60%-70% of the observational recordings are read [11].

B. Non-Monotonic Reasoning

Non-Monotonic Reasoning (NMR) refers to a certain deterministic logic that may be used to determine a solution given several inputs. NMR operates by developing a number of solutions for a given problem and arrives at the likeliest conclusion from the elimination of possible solutions based on the analysis of the inputs. One such NMR logic that has been specifically created for computer science is Clausal Defeasible Logic (CDL) which utilises logical engines to prove or disprove a conclusion [12]. The key advantage of CDL is the ability to use such logic to decipher correct conclusions intelligently when given ambiguous information. For example, mammals usually do not lay eggs. However, an echidna is a

rare specie of animal in that it is both a mammal and it lays eggs [13]. To reflect this scenario, we develop a logic map as the one pictured in Figure 1 which utilises a ‘‘Priority Relation’’ meaning that the entity connected by the arc to another and is on the clockwise of another entity will gain precedence over the other. This is utilised to give priority to a specific entity rather than a generic entity, such as echidnas and mammals respectively.

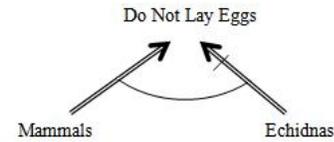


Fig. 1. The Clausal Defeasible Logic map which states that although Echidnas are Mammals, they also lay eggs [13].

To computationally represent this information, a programming language known as ‘‘Decisive Programming Language’’ (DPL) had been proposed [14]. Five symbols are utilised in this language to represent all the information that is needed for the logic engine to operate. The first symbol is the ‘‘ \rightarrow ’’ symbol that dictates a strict rule statement containing absolutely no ambiguity. The second symbol is the Defeasible Rule Symbol ‘‘ \Rightarrow ’’ that represents a plausible relation in which the statement is generally true but may be proved under certain circumstances. The ‘‘ \rightsquigarrow ’’ is utilised as a Warning Rule which states that the former entity may not disprove the latter. To represent the priority of one rule over another, the ‘‘ $>$ ’’ symbol is utilised. For example, in the instance of $A > B$, entity A is given priority over entity B. The last symbol that is used extensively is the ‘‘ \sim ’’ symbol which negates the entity it is positioned in front of. The following set of rules is used to represent the previous scenario about mammals laying eggs using DPL:

Echidna(x) \rightarrow Mammals(x) .
R1: Mammals(x) \Rightarrow Do Not Lay Eggs(x) .
R2: Echidna(x) $\Rightarrow \sim$ Do Not Lay Eggs(x) .
R2 $>$ R1 .

Additionally, CDL provides different formulae of proof engine which represent the level of confidence utilised when proving or disproving a conclusion. These formulae contain algorithms named μ , π and β . μ refers to the strongest algorithm in which only factual statements are utilised to prove the conclusion. The π algorithm allows ambiguously propagated conclusions to prove a theory whereas the β algorithm blocks any ambiguity from being utilised to discover a conclusion. This logic has already been implemented in robot soccer playing dogs and an elderly care alarm system to determine the correct action given the various inputs from each situation [13].

III. RELATED WORK AND MOTIVATION

To properly investigate our field of study we have compiled an analysis of already existing methodologies and underlying motivations as to why this research is important. We have identified three main categories of related work, the physical, middleware and deferred solutions. From this related work we have identified flaws within the current state-of-the-art technology that will not allow for the highest possible cleaning rate that may be obtainable with further intelligent correction.

A. Related Work

There have been several methodologies proposed in literature to clean up the ambiguous information ascertained from RFID readers. We have divided this into three main categories: *Physical Solutions*, in which the hardware is altered to improve the reading rate; *Middleware Solutions*, where the middleware is given algorithms to correct the data before passing it onto the database; and *Deferred Solutions* that incorporate intelligent techniques, which correct the data captured within the data storage. Several dominant physical solutions include the use of the positioning of the tags on the items [15], multiple tags or read cycles [16][10] and insulated padding to prevent metallic interference [17].

With regards to middleware solutions, the majority of the proposed methods are based on filtering the incoming sensor readings. Some of these include a smoothing window [18] and anti-collision protocols [19]. There have also been statistical approximation methods have been used primarily for sensors but they may be also utilised for RFID applications [20]. Deferred solutions correct the processed data at a later stage of the data capture cycle, usually taking advantage of past and future observations to eliminate anomalies. Several of these methodologies that have been proposed in literature include a rule-based application [21], probabilistic inference [22] and probabilistic high level event transformations [6]. In previous work, we have investigated the effect of a Plausible Logic cleaning algorithm [23], Deferred Bayesian Network [24] and a Bayesian Network that has been trained by a Genetic Algorithm [25].

B. Motivation

We believe that there are still fatal flaws within the state-of-the-art work that may allow anomalies to prevail within the storage system and that this is sufficient motivation to continue research in this field of study. We summarise the flaws with other techniques within the following:

- **Physical Solutions:** Although physical solutions may solve several causes of missed readings, there will be various scenarios in which the solution remains ineffective and will not be able to record every observation.
- **Middleware Solutions:** The filtration of the recordings will stop anomalies such as duplicate and wrong data. However we believe that there is not enough recordings stored so that an analysis can be used to restore the data.
- **Deferred Solutions:** The rule based and event transformation software have a flaw in that without a high level

intelligence to differentiate which rule gains higher precedence, the anomalies may remain. Also, transforming low level observations into high level events will undoubtedly reduce the accuracy of the data.

IV. PROPOSED METHODOLOGY

In this work, we propose the use of both data analysis algorithms and *Clausal Defeasible Logic* (CDL) engines to discover crucial analytical information and ascertain the most likely observations to load into the database. The intended architecture of our system may be viewed in Figure 2. We will then provide a comprehensive summary of the data analysis procedures utilised to find the analytical information and the CDL engines which are harnessed to determine the most correct observational recordings. Finally, we then provide an analysis of the advantages and disadvantages regarding the implementation of our proposed methodology.

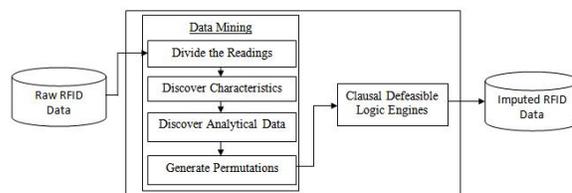


Fig. 2. The system architecture of our proposed system. Raw RFID data is streamed into the data analysis section in which it is divided, analysed and permutations are generated. This analysed information is then passed to the Clausal Defeasible Logic engines which determine the likeliest imputed variables to be stored back into the database.

A. Scenario Configuration

The schema we have chosen to store all RFID data is the “Data Model for RFID Applications” (DMRA) architecture discussed by [26]. DMRA has been implemented within the Siemens RFID Middleware to store the data. Specifically in DRMA, we have populated and used the following table:

```

READER (Reader_epc, Name, Description)
OBJECT (Epc, Name, Description)
LOCATION (Location_id, Name, Owner)
READERLOCATION (Reader_epc, Location_id)
OBSERVATION (Reader_epc, Value, Timestamp)
  
```

The ideal scenario for which we have intended to develop a system is a static RFID-enabled environment such as a supply chain warehouse that tags the items to be tracked until the departure from the facility. It is important that the scenario includes a static environment as the system hinges on the use of map data which stores two readers that are close to each other. To reflect this, we have added an additional table into the DRMA as the following:

```

MAPDATA (Reader1_epc, Reader2_epc)
  
```

Additionally, it is extremely crucial in our scenario for the readers to interrogate the tags periodically. By this, we mean

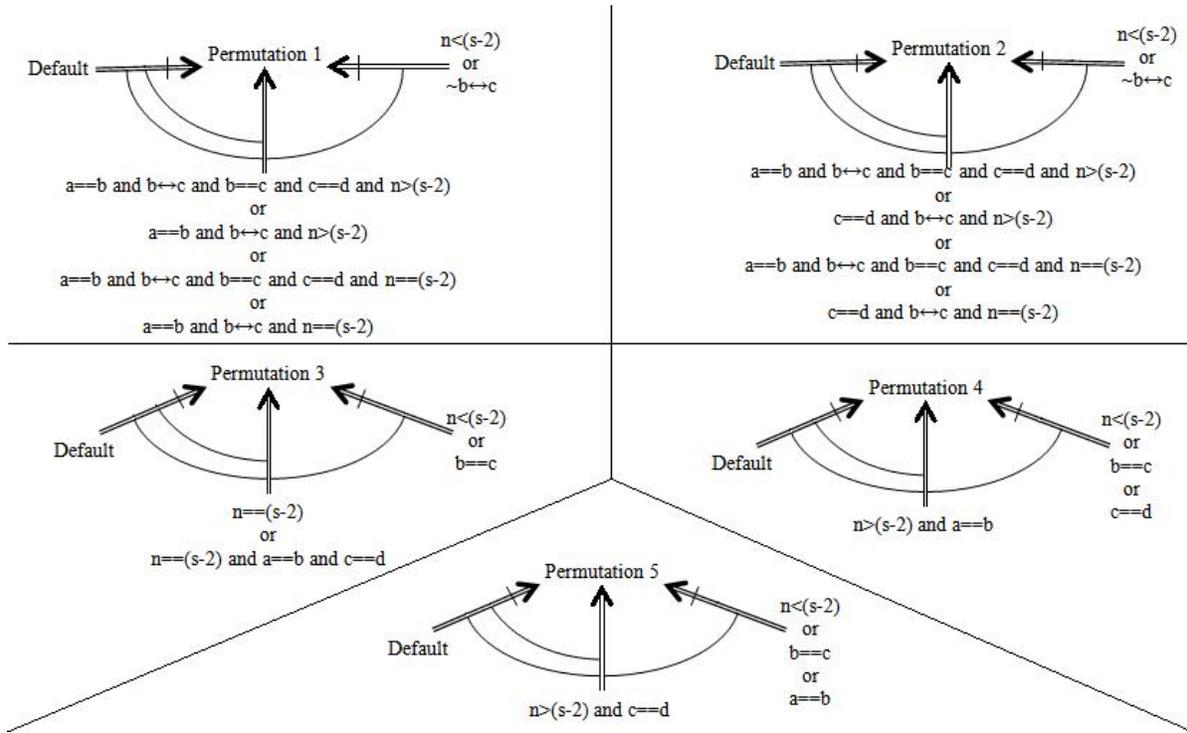


Fig. 3. The CDL engine maps utilised for the five different Permutations.

that the readers will emit the scan for the tags continuously at certain time intervals. It is important that the exact time between reads is less than the amount of time it would take a tagged item to move from one reader's range to the next. This is due to the fact that we are assuming that an item can only reach another reader's range in one missed reading. With this in mind, we can increase the accuracy of the imputed values which we generate.

B. Data Analysis

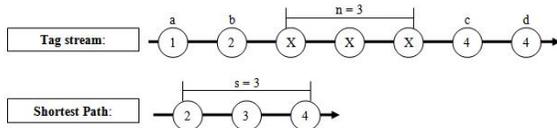


Fig. 4. An example of how the system discovers the analytical variables. In this example, the map data would state that: $1 \leftrightarrow 2$, $2 \leftrightarrow 3$ and $3 \leftrightarrow 4$. The main goal of this stage of the program is to extract the variables of a, b, c, d and n from the stream data then generate the shortest path and find the value of s. In this example, a, b, c, d, n and s would equal 1, 2, 4, 4, 3 and 3 respectively.

The data analysis algorithm has four main processes it must complete before passing the resulting information onto the *CDL Imputation engines*. The first process is to take the raw RFID readings and divide up the readings into streams of data which will only examine one tag at a time. From these streams and under the assumption that the readers record all tagged items at the same location periodically, we can discover missing gaps of knowledge as seen in Figure 4. Once a gap is found within the data, the second process is initiated in which

the algorithm attempts to find certain aspects of the missing data. The algorithm will specifically search for the reader recorded two time periods before the missed readings (a) and the reader value just before the missing data (b). The data will then attempt to find the reader values one (c) and two (d) time periods after the missing data. After these values are found, the algorithm finds the shortest path between readers b and c utilising the map data, determines how many observations values are missing (n) and finds how many observations are in the shortest path (s).

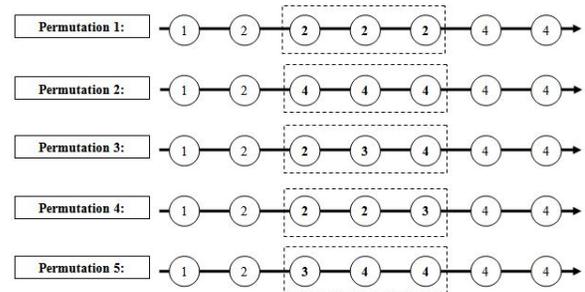


Fig. 5. The different Permutations that would be generated from the analysis data. The values inside the dotted lines reflect the imputed variables. Permutation 1 substitutes all missing values with reader b; Permutation 2 will substitute all missing values with reader c; Permutation 3 inserts the shortest path into the middle of the gap; Permutation 4 inserts the shortest path into the latter part of the gap and Permutation 5 inserts the shortest path into the former part of the gap.

After these six values are found from the gap of knowledge, the data analysis algorithm then performs its third process

to transform the data into meaningful information. It does this by finding the true or false validity of the following mathematical operations: “ $a=b$ ”, “ $b\leftrightarrow c$ ”, “ $b=c$ ”, “ $d=c$ ”, “ $n=(s-2)$ ”, “ $n>(s-2)$ ” and “ $n<(s-2)$ ”. The reason as to why we always subtract 2 from the value s is that the shortest path will also include the values of readers b and c . Also, the \leftrightarrow dictates that the readers on opposite ends of the symbol will be close to each other according to the map data. The final process of the data analysis algorithm is to generate the five permutations seen in Figure 5 to be inserted back into the data storage in place of the missing data. The first permutation substitutes each missed observation with the value of reader b whereas the second permutation substitutes the value of reader c in the missing data’s stead. The third permutation will take the shortest path and insert it as close to the middle of the gap as possible and fill the surrounding values if any with the values of readers b and c on the left and right side respectively. The fourth permutation will take the shortest path and place it on the latter part of the gap substituting the value of reader b any additional missing values on the left side. The fifth permutation is the mirror opposite of the fourth in that it will insert the shortest path to the former part of the gap and substitute all other missing values with the reader value of c . After these processes have been completed, both the permutations and the analytical data is passed into the Clausal Defeasible Logic engines.

C. Clausal Defeasible Logic Imputation

To correctly determine which permutation of values is the most likely we have employed the use of five *Clausal Defeasible Logic* engines whose configuration maps may be viewed in Figures 3. We created a logic engine for each permutation to either award a true or false value to each said imputed values. These logic engines specifically seek out the relevant information which was determined in the data analysis phase. For example, “Permutation 3” is false by default. However, if it is found that the “ $n=(s-2)$ ” value or all three of “ $n=(s-2)$ ”, “ $a=b$ ” and “ $c=d$ ” are true, “Permutation 3” will be set true. If it is later determined that either “ $n<(s-2)$ ” or “ $b=c$ ” is true, “Permutation 3” will finally be set to false. After the engines have determined if each permutation is true or false, we then apply a which states that the order of precedence for the different data sets which is that: “Permutation 3” > “Permutation 1” > “Permutation 2” > “Permutation 4” > “Permutation 5”. For example, if only Permutations 1 and 5 are found true, we will substitute the values generated by Permutation 1 into the database as we believe that this data set has a higher chance of being correct. If no permutations are found true by the engines, we employ “Permutation 3” as the default imputation.

V. EXPERIMENTAL RESULTS AND ANALYSIS

As seen in Figure 6, we conducted experimentation upon the Non-Monotonic Reasoning engine to find that the system cleans at an average rate of approximately 87.7%. Attention should be drawn to the fact that both the μ (Mu) and β

(Beta) formulas scored close to the same amount of cleaning consequently allowing the π (Pi) algorithm to obtain the highest cleaning rate. Also of interest is that the algorithm π algorithm performed the optimally when having to correct 5000 test cases and that the μ and β performed the worst when attempting to clean 2000 test cases. The system has been written in C++ and compiled using Microsoft Visual Studio 6.0 and implemented on a Windows XP machine with Service Pack 3 Intel(R) Pentium (R) 4 CPU 2.79GHz with 2.00 GB of RAM.

We have identified three core benefits from our observations of the methodology. The first advantage is that our methodology will be able to impute missing readings at a high level of correctness. The second benefit is that due to the nature of the imputed permutations, there is a higher chance that the imputed variables will be correct. This is especially important for it is rare in reality to find consecutive missed readings. The more common situation involves infrequent gaps of knowledge scattered throughout the data set. The third advantage we have identified is the use of a novel reasoning system which may determine correct outputs when given ambiguous inputs.

VI. CONCLUSION

Throughout this work we have investigated the crucial factor of missing observational data with RFID systems which prevent the wide scale adoption of the passive tag technology. We have then summarised the current state-of-the-art applications already employed by the research community and described our proposed methodology to help eliminate missing data anomalies. In summary, we:

- Briefly outlined the missing RFID data problem and non-monotonic reasoning for the reader.
- Provided a survey and classified the various methodologies that have been proposed in literature to solve missed readings such as physical, middleware and deferred solutions.
- Proposed unique methodology that utilises intelligent data analysis and a non-monotonic reasoning engine to restore missing data.
- Provided experimentation and analysis to illustrate the advantages of utilising our technique with .

With regards to the future work it would be interesting to compare the accuracy and cost of this method with a *Bayesian Network*, which we have previously established as a viable restoration algorithm. Also, to investigate and compare other classification techniques such as *Artificial Neural Networks* with our algorithm . Finally, we believe it would be worthy to create a data analysis technique coupled with non-monotonic reasoning to eliminate the other anomalies associated with RFID: wrong and duplicate data.

ACKNOWLEDGMENT

The authors would like to thank René Hexel, David Billington and Andrew Rock for their helpful comments during the earlier stages of this work. This research is partly sponsored by ARC (Australian Research Council) grant no DP0557303.

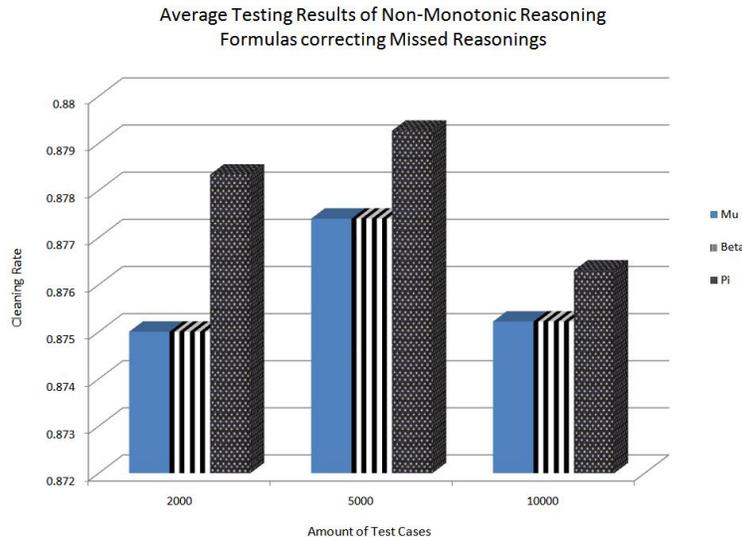


Fig. 6. The cleaning rate of the different Non-Monotonic Reasoning Formulas correcting various amounts of test cases. The solid, striped and dotted bars reflect the μ (Mu), β (Beta) and π (Pi) formulae respectively.

REFERENCES

- [1] P. Harrop, "RFID in the Postal Service [online]," MoreRFID, October 2005, available from: <http://www.mail-archive.com/medianews@twiar.org/msg05713.html> [Accessed: 12th June 2008].
- [2] J. Collins, "Boeing Outlines Tagging Timetable [online]," RFID Journal, June 2004, available from: <http://www.rfidjournal.com/article/view/985/1/1> [Accessed: 5th November 2008].
- [3] C. Swedberg, "Hospital Uses RFID for Surgical Patients [online]," RFID Journal, July 2005, available from: <http://www.rfidjournal.com/article/articleview/1714/1/1/> [Accessed: 5th June 2008].
- [4] S. S. Chawathe, V. Krishnamurthy, S. Ramachandran, and S. E. Sarma, "Managing RFID Data," in *VLDB*, 2004, pp. 1189–1195.
- [5] SCDigest Editorial Staff, "RFID News: The Five-Cent is Here, the Five-Cent Tag is Here! Well, Almost [online]," Supply Chain Digest, Jan 2009, available from: <http://www.scdigest.com/assets/newsviews/09-01-27-2.pdf> [Accessed: 24th July 2009].
- [6] N. Khoussainova, M. Balazinska, and D. Suciu, "Probabilistic RFID Data Management," *UW CSE Technical Report UW-CSE-07-03-01*, March 2007.
- [7] M. Raskino, J. Fenn, and A. Lenden, "Extracting Value From the Massively Connected World of 2015," Gartner Research, Tech. Rep. G00125949, April 2005.
- [8] F. Wang and P. Liu, "Temporal Management of RFID Data," in *VLDB*, 2005, pp. 1128–1139.
- [9] B. Carbutar, M. K. Ramanathan, M. Koyuturk, C. Hoffmann, and A. Grama, "Redundant Reader Elimination in RFID Systems," in *SECON*, 2005.
- [10] Y. Bai, F. Wang, and P. Liu, "Efficiently Filtering RFID Data Streams," in *CleanDB*, 2006.
- [11] C. Floerkemeier and M. Lampe, "Issues with RFID usage in ubiquitous computing applications," in *Pervasive Computing: Second International Conference, PERSVASIVE 2004*, A. Ferscha and F. Mattern, Eds., no. 3001. Linz/Vienna, Austria: Springer-Verlag, apr 2004, pp. 188–193.
- [12] G. Antoniou, D. Billington, G. Governatori, and M. J. Maher, "Embedding Defeasible Logic into Logic Programming," *Theory and Practice of Logic Programming*, vol. 6, no. 6, pp. 703–735, 2006.
- [13] D. Billington, "An Introduction to Clausal Defeasible Logic [online]," David Billington's Home Page, Aug 2007, available from: <http://www.cit.gu.edu.au/~db/research.pdf> [Accessed: 3rd July 2008].
- [14] D. Billington, V. Estivill-Castro, R. Hexel, and A. Rock, "Non-monotonic Reasoning for Localisation in RoboCup," in *Proceedings of the 2005 Australasian Conference on Robotics and Automation*, C. Sammut, Ed., December 2005.
- [15] A. Rahmati, L. Zhong, M. Hiltunen, and R. Jana, "Reliability Techniques for RFID-Based Object Tracking Applications," in *DSN '07: Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 113–118.
- [16] H. Vogt, "Efficient Object Identification with Passive RFID Tags," in *Pervasive '02: Proceedings of the First International Conference on Pervasive Computing*. London, UK: Springer-Verlag, 2002, pp. 98–113.
- [17] Emerson & Cuming Microwave Products, "Emerson & Cuming Microwave Products – ECCOPAD [online]," Emerson & Cuming Microwave Products, 2008, available from: <http://www.eccosorb.com/europe/english/page/66/eccopad> [Accessed: 16th August 2008].
- [18] S. R. Jeffery, M. N. Garofalakis, and M. J. Franklin, "Adaptive Cleaning for RFID Data Streams," in *VLDB*, 2006, pp. 163–174.
- [19] D.-H. Shih, P.-L. Sun, D. C. Yen, and S.-M. Huang, "Taxonomy and Survey of RFID Anti-Collision Protocols," *Computer Communications*, vol. 29, no. 11, pp. 2150–2166, 2006.
- [20] A. Deshpande, C. Guestrin, S. Madden, J. M. Hellerstein, and W. Hong, "Model-driven data acquisition in sensor networks," in *VLDB*, 2004, pp. 588–599.
- [21] J. Rao, S. Doraiswamy, H. Thakkar, and L. S. Colby, "A Deferred Cleansing Method for RFID Data Analytics," in *VLDB*, 2006, pp. 175–186.
- [22] R. Cocci, T. Tran, Y. Diao, and P. J. Shenoy, "Efficient Data Interpretation and Compression over RFID Streams," in *ICDE*. IEEE, 2008, pp. 1445–1447.
- [23] P. Darcy, B. Stantic, and R. Derakhshan, "Correcting Stored RFID Data with Non-Monotonic Reasoning," *Principles and Applications in Information Systems and Technology (PAIST)*, vol. 1, no. 1, pp. 65–77, 2007.
- [24] P. Darcy, B. Stantic, and A. Sattar, "Improving the Quality of RFID Data by Utilising a Bayesian Network Cleaning Method," in *Proceedings of the IASTED International Conference Artificial Intelligence and Applications (AIA 2009)*, 2009, pp. 94–99.
- [25] P. Darcy, B. Stantic, and A. Sattar, "Augmenting a Deferred Bayesian Network with a Genetic Algorithm to Correct Missed RFID Readings," in *Malaysian Joint Conference on Artificial Intelligence (MJCAI 2009)*, 2009, pp. 106–115.
- [26] S. Liu, F. Wang, and P. Liu, "A Temporal RFID Data Model for Querying Physical Objects," *TimeCenter*, Tech. Rep. TR-88, 2007.