

Distributed Coordination of Mobile Robots Using RFID Technology

TANEL TAMMET¹, JÜRI VAIN¹, ALAR KUUSIK²

¹Department of Computer Science, Tallinn University of Technology
Ehitajate tee 5 Tallinn 19086,
ESTONIA

²ELIKO Technology Competence Centre,
Ehitajate tee 5 Tallinn 19086,
ESTONIA

Abstract: Radio Frequency Identification is a promising technology for realizing distributed knowledge databases of the future. The proposed RFID database solution provides a cheap and feasible solution for indoor navigation, long term information storage and exchange between autonomous mobile devices and/or humans. For device interoperability it is proposed in the paper to organize data stored on tags using methods known in the semantic web research. The timeline scalability of the given solution is demonstrated on a cleaning robots swarm example, which is formally proved using a model checking technique.

Key-words: distributed knowledge, intelligent agent, mobile robot, formal verification, semantic web, rfid

1 Introduction

Ubiquitous computing is drawing increasing attention in distributed applications: environment and industrial process monitoring, safety inspection and many others. In all the referred cases untargeted data distribution and propagation is considered useful when the robustness of an operation is an issue, e.g., guiding a maintenance robot swarm in ambiguously defined service areas, navigating robots to the rescue and technical maintenance targets. Especially in cases where the mobile devices have limited sensing capabilities, the information delivered about inspected objects and environmental conditions helps considerably in reducing the (computational) costs of decision making and planning.

On the other hand, broadcasting untargeted information leads easily to unwanted information overhead. That sets high demands on communication throughput and on energy consumption of autonomous devices. For that reason the focus of ubiquitous networking is rather on local communication than on keeping and broadcasting large volumes of data centrally.

Locality of communication implies that data should be kept close to their “host” objects they are characterizing or with mobile agents they are needed for. When the location and object specific (e.g., identifying) information has to be actualized

and accessible in real time, it is clearly infeasible to store such data in remote and irregularly accessible databases.

The data that reflects properties of real physical objects may have a highly varying lifespan - from seconds to several years - depending on the application. For the same reason, the meaning and format of data about the same objects may diverge widely. That motivates the usage of “loose” data formats that support the scalability of heterogeneous applications better than formats optimized for specific applications.

Contemporary research on mobile devices in a ubiquitous computing environment is focused mainly on the ability of position recognition. Stationary or mobile Radio Frequency Identification (RFID) [1] antennas are used to detect the position of mobile RFID-tagged objects, e.g., RFID gates in logistics and RFID combined with motion trackers.

On the other hand, *self-positioning* approach for autonomous mobile cleaning robots is proposed in [2]. Enhanced with an RFID antenna, devices can learn their present location by detecting small RFID tags spread on the floor and adjust their behavior according to this information. An alternative idea is developed in the SmartCarpet project [3], where foil-thin RFID tags approximately 1.5 cm square are embedded in a grid pattern to the back of carpeting.

In this paper a broader view about the possibilities of applying RFID technology for

storing and propagating sparse and heterogeneous data, characteristic to mobile applications will be proposed. Our goal is to create RFID media based knowledge organizing and handling environment for variety of mobile service robot swarm [4] developed by different vendors for different applications.

The novelty of the proposed approach for handling sparse and heterogeneous data of tags lies in adapting efficient knowledge engineering techniques introduced in the semantic web research. In the case study of cleaning robots the proof of the proposed concept will be demonstrated using formal model checking. It is shown that the coordination algorithm that uses relatively simple protocol based on data stored on RFID tags is timeline scalable.

2 RFID Based Robot Coordination

The robot swarm exhibit self-organization and emergent behavior that result from *scalable communication mechanisms* and *decentralized control strategies*. Collective behavior emerges from the inter-robot interactions and the interactions of the robots with the environment. Scalability can be reached mostly by means of local communication. Local communication is usually achieved by wireless transmission systems, using radio frequency or infrared communication. It has to be underlined that, from the control efficiency point of view, access to collected (event)history database might be more effective than extensive realtime sensing.

In [5] it is pointed out that further research is needed on finding methodologies that allow for designing, and reliably predicting, swarm behavior, given the features of the individual swarm members. One has to state at once that efficiency and scalability of swarm depends equally on the design of swarm environment. It will be demonstrated below that designing swarm environment as pervasive communication media allows to simplify individual behaviors of robots to the extent that is predictable not only by using incomplete analysis techniques like simulation but also by means of complete proof techniques such as model checking. To start addressing the problem a suitable media for distributed knowledge in mobile agent systems has to be defined.

Assume the data processing is performed by robots and data storages do not need intelligence or data processing capabilities. Then following requirements apply to the data media:

- The data are easily accessible and rewritable in the sense of established

communication. In application level both requirements should be optimally satisfied with sufficient range bidirectional wireless communication. For indoor application reasonable access range is several meters.

- Sufficiently long data retention time. For example, talking about service of engineering networks one can expect that local data labels can store information about failures for tens of years. Nonvolatile type data memory is preferred due the energy source limitations.
- Sufficient amount of flexibly organized data memory that can be used in different semantic contexts. Since the usage time of data is expected to be years the data format cannot be predetermined. Also simultaneous access to shared data by multiple devices should be supported.
- The available data content is location dependent. Data or pointers to data should be physically related to the object of interest to minimize informational overhead.

RFID technologies provide suitable media for storage of distributed knowledge applicable also for mobile robot control. RFID devices can be divided into following categories: active (battery powered) and passive (electromagnetic field powered) devices; medium frequency (up to 13.56MHz) devices and ultra high frequency (UHF) devices (0.8-2.45GHz). Passive UHF RFID is the most suitable technology for realizing distributed knowledge bases. In the first place, because of optimal read-write range of up to 10m, zero standby power consumption and low price of data labels. The size of data memory, currently several hundred bytes, is expected to increase soon.

3 Semantic Context on Tags

Our core scenario assumes that:

- A set of relatively simple mobile robots (a swarm of robots) exists.
- The environment is equipped with a number of RFID tags.
- Robots are able to read and write data on the RFID tags, using tags (similarly to graffiti, whiteboard or post-it-notes) for communication.
- The robots may be tasked to either clean up the environment, look for specific objects or perform other cooperative tasks.

The efficiency of the operation of the robot swarm depends heavily on the efficiency of

communication. For example, a robot would benefit from knowing whether another robot with a similar task has been around recently. A cleaning robot may decide to proceed to another area in case another cleaning robot actually did some cleaning of the area not too long time ago.

Information on the RFID tags may be also used for external communications. The robot swarm operator may, for example, mark a tag with an instruction for robots to keep away from the area, or - vice versa - a need to perform some extra cleaning. Similarly, some tags may be connected to external reception software, so that robot reports about the progress of cleaning or search on the tags is automatically reported to the operator of the swarm.

The following issues will be considered:

1. Which principles/standards to use for encoding data on rfid tags?
2. How to conserve space on rfid tags?
3. How to convert data on rfid tags to standard xml representation of rdf?
4. How to drop/overwrite data in case the rfid tag has no more free space on it?

3.1 Tag contents as rdf triplets with a context

What should a robot - or a human giving hints to robots - write to the tag? Information like:

- Robot nr 12 has been here at 10:12 on 24. January and was actively cleaning.
- Robot nr 12 has been here at 12:10 on 26. January but did not actively clean the area.
- Robot nr 15 has been here at 14.20 on 28. January looking for tag nr 244 and did not find a tag while here.
- Robot nr 17 has been here at 14.35 on 28. January looking for tags of the product type 'book' and found three.
- It is prohibited by the operator to water plants closer than 2 meters of the tag.

Actual robot devices in the swarm, and in particular, their software systems may be widely different from each other. They are probably produced by different companies in several years. Thus it is crucial to design a communication system, i.e. the principles of data organization on the RFID tags, which would be relatively easy to use by different software developers. One should also avoid excess verbosity, i.e. data representation should be relatively compact.

First of all, it cannot be assumed that RFID tags keep data written by robots indefinitely: there is too little space for that.

On the other hand, it should be assumed that some information snippets - like instructions given by the swarm operator - should not be overwritten by robots. Hence, the queues of different priorities are needed.

The actual data content of the information snippet will vary wildly, but it is essential to find a uniform information encoding scheme in order to simplify creation of reading/writing software. The simplest uniform scheme is a subject-predicate-object triplet scheme recently popularised by the semantic web languages rdf, rdfs and owl [9,10]. Using such rdf-based languages would give several benefits. These languages have been designed for distributed agents and distributed databases from the very beginning. They contain the namespace machinery necessary for distinguishing separate languages and creating unique id's. They do also contain ontology rule machinery necessary for simple inferences, like deducing that something with a type cleaningrobot is a robot and is not a swarm operator.

Following the earlier example, one could give triplets like:

- [12, performingaction, cleaning]
- [12, notperformingaction, cleaning]
- [15, lookingfortag , 244]
- [15, notfoundtag , 244]

Using triplets will inevitably mean that recording one data item may require several triplets to be written. Take, for example, a sentence "Robot nr 15 has been here at 14.20 on 28. January looking for tag nr 244 and a did not find tag while here.". Several separate facts will be derived from the sentence:

- [15, hastype, robot]
- [15, washereattime, 14.20 on 28 January]
- [15, waslookingfortag, 244]
- [15, didnotfindtag, 244]

It is obviously clear that these four triplets have to be read together - their meaning will be lost when read separately. Hence, either a special reification machinery for encoding the context information or attaching a special id or context marker to each of them has to be used. Since the latter is a significantly simpler solution, the rdf community has been discussing (see [11]) a need for the explicit context marker for some time now. Considering our context of moving robots, the main components of the context are:

- id of the robot or operator writing the snippet
- robot clock time of writing the snippet

It has to be considered the danger of older data

items being dropped from the RFID tag, rendering dependent information items incomprehensible. It can be recommended to decorate each data snippet with full metainformation, guaranteeing comprehensibility of each snippet. Once attaching the two data items mentioned here to the data snippet, these two elements will effectively denote the context, removing the need for inventing special identifiers. Thus the previous example triples, enriched with the metainformation, will become quintets:

- [15, 14.20 28 January, 15, washereattime, 14.20 on 28 January]
- [15, 14.20 on 28 January, 15, waslookingfortag, 244]
- [15, 14.20 on 28 January, 15, didnotfindtag, 244]

The "quintet" concept is still a simplification. The object, predicate and subject symbols need a namespace/type indicator. The namespace makes the symbol globally unique, while the type indicator says that the symbol should be interpreted as a pure string value (say, address of somebody) or integer/float/date value (rdf actually employs xml schema datatypes).

3.2 Efficient rdf encoding on RFID tags

Modern passive UHF RFID tags contain approximately 256 bytes of user-writable memory. This memory is assumed to be split into segments of length between 4 and 32 bytes (see the ISO18000-6x standard).

One self-contained item of information (triple plus metainformation) would be stored in one segment. The 32 byte segment could be used for the data block, splitting it into 8 four-byte integers. The first two integers will encode the metainformation, the remaining six will encode the subject-predicate-object triple along with the namespace for each element. The namespace can be assumed to be either a string or a type indicator, indicating that the following integer value is a direct string value in the string table, a char, an integer, a float, etc.

Thus the integers of each block will have the following semantics:

1. nr of the robot
2. datetime of writing, according to the robot clock
3. nr of the namespace string of the subject
4. nr of the subject string
5. nr of the namespace string of the predicate
6. nr of the predicate string

7. nr of the namespace string of the object
8. nr of the object string

Since integers for strings are proposed, one will have to devise a suitable encoding scheme used by all the robots. The suggested scheme employs several string tables, which have to be augmented and read by robots before starting the work in the swarm. Each robot will have to write all the symbols/strings employed by its program in the rdf context into the table. It will also have to read the encodings of other robots, except for the robots it will not communicate with anyway. There are three tables:

- **robot table** contains pairs <integer,unique_id_of_robot> where the mac address of the ethernet or wifi card of the robot may be used as a unique id. This table is used for obtaining the full robot id from the first integer in the rfid data block.
- **string table** contains triples <robot nr, string nr, string> where the robot nr is the first element of some pair in the robot table and the string nr is an element between and incl. 3rd and 8th integer of the data block. This table is used for translating namespace and object, predicate, subject numbers into full strings.
- **robot mapping table** contains pairs <robot nr in the data block, mapped robot nr in the string table> where the mapped robot nr - if present - has to be used for the string table instead of the robot nr in the data block. The table allows to share the string table contents for a large number of robots, avoiding duplicate numeration of the same string for several robots.

In order to map a namespace nr into the string, one has to take the string from a corresponding string table triple, using the mapped robot of the data block robot nr as the first element of the triple.

Differently from the namespace numbers, the subject, predicate or object nr is not necessarily mapped to a symbol string. Instead, the corresponding number in the data block may encode a proper (non-symbol) string, integer, float, date or similar. Our proposal is to:

- Use negative numbers in the data block to indicate a direct type instead of a string table element.
- Use direct encoding for direct string values and the following datatypes as used in the C language and the standard libraries: string, char, int, float, datetime, date, time. Corresponding integers are -1, -2, ..., -6. The types will be eventually converted to

type-decorated strings according to the XML schema datatypes.

3.3 Converting RFID data blocks to rdf in xml

It will be convenient for the robot software to use RFID data as rdf in the standard xml format. This will allow the rdf software to use the existing rule languages - rdfs, owl and others - as well as the existing software for handling rdf. The process to convert RFID data to rdf in xml contains two major steps:

- Convert the integer representation of rfid data blocks to strings and direct values, using the algorithms described in the previous chapter.
- Convert the data obtained to the proper rdf xml format.

The basic idea of the latter step is using the reification machinery of rdf (see [9]), allowing us to treat each rfid data block as a separate item containing subject-predicate-object triple as well as the metainformation: unique robot id and the datetime of the robot. An unique id string of the event from the latter two will be constructed. Consider an example quintet:

[15, 14.20 on 28 January, 15, lookingfortag, 244].

This quintet should (assuming the namespaces are given as below) be translated to the following proper rdf (although several strings are presented as shorthand):

```
<rdf:Description
xmlns="http://www.w3.org/..."
xmlns:rdf="http://www.w3.org/..."
xmlns:rob="http://robots.org"
xmlns:r1="http://example.org"
xml:base="http://robospace.org/event"
rdf:ID="123-233-201at2006-01-28T14:20">
  <rdf:type>Statement</rdf:type>
  <rob:robnr>123-233-201</rob:robnr>
  <rob:dtime rdf:datatype="&xsd;dateTime">
    2006-01-28T14:20
  </rob:dtime>
  <rdf:subject>
    rob:123-233-201
  </rdf:subject>
  <rdf:predicate>
    r1:lookingfortag
  </rdf:predicate>
  <rdf:object rdf:datatype="&xsd;decimal">
    244
  </rdf:object>
</rdf:Description>
```

3.4 Drop/overwrite data on RFID

The "typical" RFID tag has space to contain about eight data items. It is obvious that robots will have to drop old data blocks on tags, in order to write new ones. The algorithm for dropping old data blocks should follow two main principles:

- The robot software should contain information about privileged robots (might be people) whose information must be never dropped.
- For all the other robots it will make sense to drop the oldest data blocks first, essentially making the RFID tag behave like a queue.

It is clear that in addition to the simple two-level queue the robot software might be constructed to contain complex priorities of robots and/or data categories. However, other robot types will probably know nothing about these priorities and the additional complexities will make behaviour of the swarm very hard to predict and analyze.

4 A case study: Swarm of Cleaning Robots

As an example of using RFID tags for distributed control of robot swarms the operation of cleaning robots (e.g. iRobot) in a hypothetical factory environment is described. The properties of proposed planning algorithm are formally proved.

4.1 Informal description

The goal is to coordinate cleaning robots so that they can operate simultaneously without disturbing manufacturing process and each other when cleaning the area. Commercially available cleaning robots have typically sonars for obstacle avoidance but they cannot determine if the area is already cleaned or not. In our case study the robots are equipped additionally with RFID read-write devices (R/W-devices) to read/write the cleaning information from RFID tags. Passive RFID tags are attached to floor, material and waste containers. Human operator moving in the area also carries a label. Operating steps of the robot are:

1. Find a landmark.
2. Determine if nearby area needs cleaning.
3. Detect if area is available for cleaning.
4. Proceed with cleaning.
5. Stop when there is a moving object on the route
6. Write cleaning report on the tag.

Robots distribute essential information to coordinate cleaning operations by communicating

via RFID tags only. No human action is required to determine cleaning areas or start/stop the cleaning process.

4.2 Assumptions of the cleaning algorithm

At first, assumptions needed to determine robot routes between cleaning areas are clarified. To avoid duplication and/or incomplete cleaning the cleaning area is divided into possibly intersecting zones where each zone is identifiable by RFID tags attached to the objects of that zone (see Fig. 1).

The objects in the zone may be moving or steady. Assume that the tags of steady objects of a zone allow defining some reference area (preferably a center) of that zone. The robot switching between zones means here its motions from the reference area of one zone to that of another zone. In our case study it suffices to assume that RFID tags of the zone have following obligatory data fields: ID of the zone, time-stamp of the last cleaning, flag “occupied” showing whether the zone is currently occupied by some robot or is free for new cleaning.

It is assumed that the R/W-device attached to the robot can simultaneously read and write tags that are in its visibility range. Tags are placed evenly in the room. At each tag the reader can see at least 4 other tags when in the middle of room, 3 tags when in the border (close to the wall), and 2 tags when at the corner zone of the room (see Fig. 1).

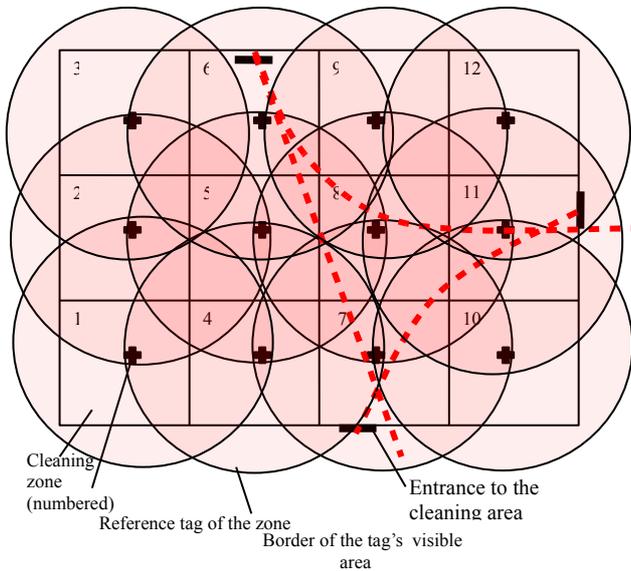


Fig. 1 Visibility of cleaning zone identification tags

Second, assumptions about soiling dynamics and robot performance are introduced. Cleaning zones may soil with different rate, which is

measured in percents of usability change per time unit: 0 % corresponds to the clean room and 100 % is the level where the room is not usable for given application anymore. The threshold TR of acceptable (for given context) soiling level may range within the interval [0; 100).

There are two phases and two phase switching events describing the cleaning dynamics:

- During the soiling phase the soiling rate is constant and specific to the zone.
- In the cleaning phase the soiling level is defined approximately by an open interval $(0, l)$, where l is estimated level of soiling in the beginning of cleaning phase.
- At “start” of cleaning the increase of soiling alters with decrease of soiling.
- Cleaning terminates at the soiling level 0 and soiling phase starts again.

Necessary performance assumption for solving the cleaning problem is that the average soiling rate is less or equal than the average cleaning rate, resp. $V_s \leq V_c$ where V_s, V_c are measured in $(\% \cdot \text{area unit})/\text{time unit}$. Also the duration of moving robots between zones is taken into account in V_c . Cleaning rate V_c does not depend on the shape of the zone. For simplicity it is assumed that the areas of cleaning zones and their cleaning times (denoted d_z) respectively are equal for all zones. Obstacle avoidance and cleaning trajectory planning problems are not considered explicitly in this setting.

4.3 RFID placement and the cleaning algorithm

The visibility range r of R/W-devices is 4 meters. For simplicity it is assumed the circular shape of its visibility area of about 50 m². Since in all points of the room at least one tag has to be visible the distance between neighbor tags cannot exceed the limit of 4 m. One possible placement of tags is depicted in Fig. 1.

Cleaning of one zone is an atomic action, i.e., once started the robot does not interrupt or take another zone before it has completed with given zone.

Only one robot at a time can clean a zone. To block others clean the same zone the sign “occupied” is written in the zone’s tag.

After completion of cleaning phase the robot deletes the “occupied” sign and writes a new time-stamp in this tag. For planning further activities the robot checks the time-stamps of tags visible from the reference area of its location zone. It finds the zone(s) with oldest time-stamp(s) that are not

occupied by some other robot yet. Then it moves to the reference area of the target zone and repeats the inspection cycle there again. If the tag of robot's current location has the oldest time-stamp among those visible around, the zone is chosen for cleaning next.

4.4 The formal model

The task of showing feasibility of given in Section 4.4 technical solution consists of two related problems:

- Modeling problem: to apply the model checking technique for proving properties of the coordination algorithm of Section 4.4 construct the behavioral model of the cleaning system where soiling process, the area exploration algorithm of a robot, the number of cleaning robots, performance parameters V_c, V_s, l , and the partition of room into zones are explicitly represented.
- Verification problem: given the model of cleaning system prove that the soiling level is always kept below the critical threshold TR.

The model of the robot swarm is constructed using extended timed automata [6]. Four basic model templates are used: SOILING, MAP, AGENT, COMMUNICATION (Fig. 2). Template SOILING represents soiling as discrete process with time step parameter $time_step = 15$ (time units). The soiling rates of zones indexed by $1, \dots, n$ ($n = 12$) may be different for each zone and are parameterized using constants $dif1, \dots, dif12$. In Fig. 1 the most intensively soiling zones are those remaining on operator routes denoted by dotted lines between entrances. Thus, the soiling parameters are $dif1, \dots, dif4, dif9, dif10, dif12$ are equal to 5 and $dif5, \dots, dif8, dif11$ are equal to 10 (% per time unit). The current values of soiling levels are stored in the vector tag each element of which corresponds to a zone.

The behavior of a cleaning robot is modeled using templates MAP, AGENT, and COMMUNICATION (Fig. 2). Locations (meta-states) of the template MAP denote cleaning zones. Since the local planning algorithm for choosing the zone to be cleaned next uses only directions of visible tags there are transitions between only those model locations that correspond to neighbor zones.

Movement through zones is guided by variables next. For each robot the variable next is valuated as the result of its planning loop in (corresponding to that robot) template AGENT. Template AGENT has locations Cleaning, Decide_action, Moving,

In_the_zone that model respectively cleaning, planning next target zone, moving from one zone to another, and being in the reference area of the zone. Template COMMUNICATION models activities of the robot when it polls tags visible at its position. When polling, those tags where the flag Occupied is up, are ignored.

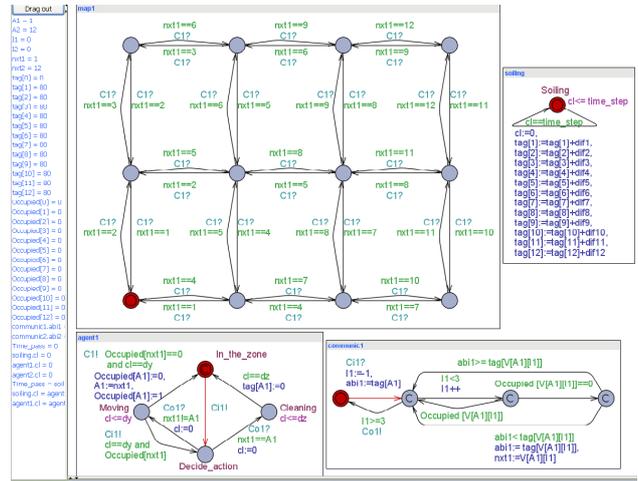


Fig. 2 Formal model of the cleaning system

4.5 Correctness of the swarm coordination algorithm

Correctness of the proposed solution is verified using model checking technique [7] and Uppaal tool [8]. The goal is to prove that given the model of cleaning system the soiling level is always kept below the critical threshold TR. In our example $TR = 51$. To prove the scalability of given property in time ("always" part of the assertion) the proof will be split into two steps:

First, it will be shown that from the state where the soiling level of all zones is over the threshold TR, e.g., 80 %, the state where the soiling level of all zones is less than TR (e.g., $TS = 30$ %) is always reachable. For that the initial value of vector $tag = \{80, \dots, 80\}$ is defined satisfiability of this reachability property will be proven by running Uppaal query: $A \langle\langle tag[1] < TS \text{ and } tag[2] < TS \text{ and } tag[3] < TS \text{ and } tag[4] < TS \text{ and } tag[5] < TS \text{ and } tag[6] < TS \text{ and } tag[7] < TS \text{ and } tag[8] < TS \text{ and } tag[9] < TS \text{ and } tag[10] < TS \text{ and } tag[11] < TS \text{ and } tag[12] < TS \rangle\rangle$ with constant $TS = 30$ %.

Second, safety assertion has to be proved. The safety assertion states that given the set of all cleaning zones with soiling level less than some constant TS where $TS < TR$ the soiling level is always kept

below the threshold TR. Having sufficiently long time horizon (by enumerative model checking one can prove only bounded time properties), e.g., TH = 240 (time units) for our case study, satisfiability of given property is proved by following query: $A[(\text{tag}[1] < \text{TR} \text{ and } \text{tag}[2] < \text{TR} \text{ and } \text{tag}[3] < \text{TR} \text{ and } \text{tag}[4] < \text{TR} \text{ and } \text{tag}[5] < \text{TR} \text{ and } \text{tag}[6] < \text{TR} \text{ and } \text{tag}[7] < \text{TR} \text{ and } \text{tag}[8] < \text{TR} \text{ and } \text{tag}[9] < \text{TR} \text{ and } \text{tag}[10] < \text{TR} \text{ and } \text{tag}[11] < \text{TR} \text{ and } \text{tag}[12] < \text{TR} \text{ and } \text{gclock} < \text{TH})$ where gclock denotes a clock variable. Time horizon TH = 240 time units means that each intensively soiling zone has been cleaned at least 5 times and less intensively soiling zones 3 times.

Experiments with the model of Fig. 2 having 12 zones cleaned by two robots with soiling constants TS = 30 % , TR = 51 % and initial soiling level 80 % show that both assertions are satisfiable. The model checking procedure with X86-based PC having 2 GHz processor and 1 GB memory takes with reachability query only few seconds and with safety assertion about 197 seconds.

5 Conclusion

One of promising application of Radio Frequency Identification technology is distributed knowledge based control of mobile devices. In this paper it was proposed to organize data on RFID tags using methods known in semantic web research to simplify interoperability of different devices. The timeline scalability of given solution is demonstrated on a cleaning robots swarm example and proved formally using model checking technique. Our further research is related with studying spatial scalability of RFID tag-based coordination and communication algorithms for large robot swarms and developing corresponding proof method for that.

Acknowledgements:

This research was supported by Enterprise Estonia and by Estonian Science Foundation grant No 5775.

References:

- [1] Finkenzeller, K., RFID Handbook: Fundamentals and Applications in Contactless Smart Cards and Identification. John Wiley and Sons, 2003.
- [2] S. Domnitcheva. Smart Vacuum Cleaner – An Autonomous Location-Aware Cleaning Device.
- [3] T.Staedter. Intelligent Carpet Directs Robot Vacuum. *Discovery News*.

- <http://dsc.discovery.com/news/briefs/20050613/smartcarpet.html>
- [4] <http://www.swarm-robotics.org/>
- [5] http://en.wikipedia.org/wiki/Swarm_intelligence
- [6] R. Alur and D.L. Dill, A theory of timed automata, *Theoretical Computer Science* 126:1(183--235), 1994.
- [7] Edmund M. Clarke, J., Grumberg, O., Peled, D.A.: Model checking. MIT Press, Cambridge, MA, USA (1999)
- [8] Amnell, T., Behrmann, G., Bengtsson, J., D'Argenio, P.R., David, A., Fehnker, A., Hune, T., Jeannet, B., Larsen, K.G., Möller, M.O., Pettersson, P., Weise, C., Yi, W.: Uppaal - Now, Next, and Future. In Cassez, F., Jard, C., Rozoy, B., Ryan, M., eds.: Modelling and Verification of Parallel Processes. Number 2067 in Lecture Notes in Computer Science Tutorial, Springer-Verlag (2001) 100–125.
- [9] Manola, F., Miller, E., eds. RDF Primer. <http://www.w3.org/TR/rdf-primer/>
- [10] Smith, M. K., Welty, C., McGuinness, D. OWL Web Ontology Language Guide. <http://www.w3.org/TR/owl-guide/>
- [11] MacGregor, R., Ko I-Y. Smith, M. K., Representing Contextualized Data using Semantic Web Tools. In *PSSSI - Practical and Scalable Semantic Systems*, Proceedings of the First International Workshop on Practical and Scalable Semantic Systems, Sanibel Island, Florida, USA, October 20, 2003