Interfacing with Next Generation Tagging and Tracking Systems for Prisons and Correctional Facilities

Catherine McFarland, Roy Sterritt, Patricia O'Hagan, Edward Hanna
University of Ulster, Core Systems
R.Sterritt@ulster.ac.uk, Patricia@coresystems.biz, Edward@coresystems.biz

Abstract

Current state of the art biometric tracking systems for Prisons and correctional facilities rely on entry points to know who is where. It is envisaged that next generation systems will be required to tag and track at the individual level. Previously the authors have presented a roadmap for such a cyber-physical system. This paper aims to provide a potential interface design to facilitate management and interaction within and to the self-managing system.

1. Introduction

Prisons and correctional facilities, particularly within Europe, have moved away from the traditional role of restrictive management and into one more akin to mentoring and re-education. This change of direction coupled with staffing shortages creates the need for security systems capable of self-management to relieve prison staff of the more fundamental tasks of managing movement and interaction and allow them to focus on the individual. Even in the US, where this has not been the focus, the legacy of high rates of imprisonment have resulted in the need for self-management systems [8].

Previous work detailed a potential design roadmap [1] to the tagging and tracking element of an existing self-managing security system for use in prisons and correctional facilities [7]. Logical processes and autonomic computing elements had been suggested to support self-management while tracking technology solutions were discussed that best uphold optimum data security.

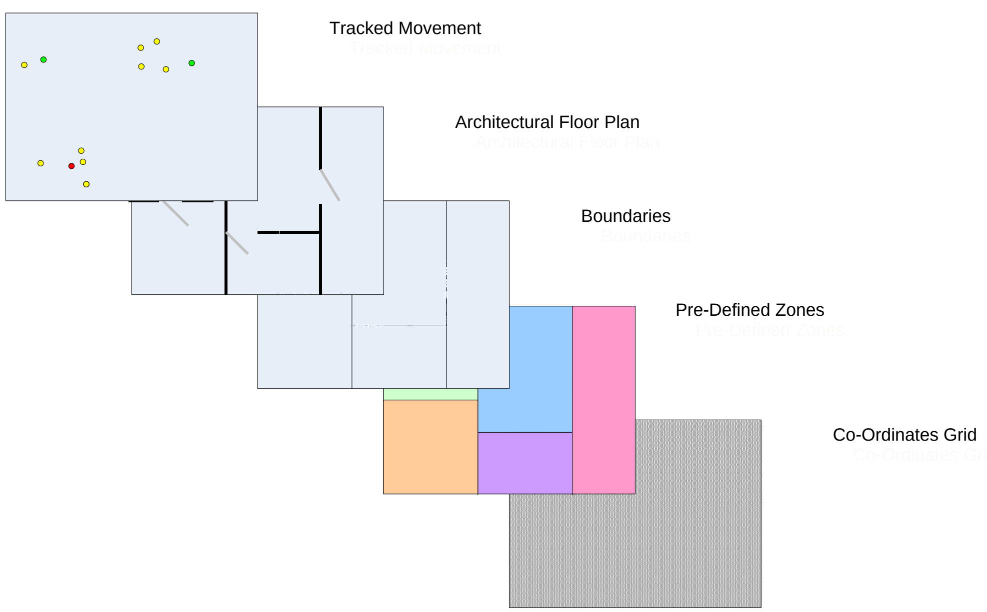

Figure 1. Software Planes

This paper extends from the original roadmap design [1] to offer a proposed user interface to work with the complex processes and capabilities previously detailed. Since the system is intended for use in secure correctional facilities some assumptions have been made as regards the hardware infrastructure and its capabilities.

While this paper's main aim is to detail the interface without changes to the original design roadmap, where consideration for the interface and its performance have led to further ideas on the background processes these have been included.

We will begin by looking at the original software planes in more detail and then looking at the two main complexities for such an interface, those being: how to represent the often complex floor plans of a correctional facility and how to represent the halo collisions [1].

2. The Software Planes

There are five conceptual software planes recognized for the interface:

- Tracked Movement
- Architectural Floor Plan
- Boundaries
- Pre-defined Zones
- Co-ordinates Grid

Each layer has a significance and function for the final operator interface based on its original role and the objects it is intended to represent.

2.1 Co-ordinates Grid

The co-ordinates grid is the base upon which all objects from the subsequent layers are defined. Co-ordinate zones are also used to calculate halo radii around tracked objects (prisoners, staff, etc.) to identify halo collisions (figure 3.) and tardy responses (figure 2.).

This layer is not visible to the user.

2.2 Pre-defined Zones

For ease of monitoring prisons can be logically split into zones. These zones may encompass an entire floor, single corridor (with adjoining cells), recreation areas, etc. Definitions of zones should fall to prison authorities as they deem most appropriate for the level of monitoring required and the capabilities of staff volume and hardware.

The concern for the interface is ease of use and clarity of information so each zone should be kept to a size that can be easily and functionally represented in a single computer screen or application window.

2.3 Boundaries

The original intention of the boundaries was to offer a layer that provided physical limitations on movements and ensure the system represented movement logically throughout the prison by detailing immovable boundaries that tracked objects would have to move around rather than through (such as walls).

These illegal movements are largely to facilitate logical processes in the backend of the system away from the operator but may also identify illegal movements possibly pointing to tag removal or inappropriate activity where the inmate wishes to confuse the system regarding their exact location. As a result, while the boundaries themselves will not be logically represented in the interface any logical violations that they may highlight, e.g. the movement of a tag suggesting the inmate has passed through a wall or secured access point, are.

Representation of such violations will appear as textual and colour-coded warnings on the screen as detailed later in the paper.

2.4 Architectural Floor Plan (AFP)

Sitting over the top of the boundaries the logical floor plan will be one of the main components of the user interface's primary view.

This will offer a true-to-life view of the zone specified by the user. While it is best to limit aesthetic detail and complexity for this view to provide good performance while ensuring the resultant hardware requirements are not financially inhibitive, the view should still be of a quality that each zone and its constituent objects are immediately recognizable. To make the system marketable the minimum hardware requirements must be kept within feasible budget limitations.

For especially complex floor plans there may come the need for a three-dimensional perspective that will necessarily increase the processing overhead of the system and so any associated hardware costs.

2.5 Tracked Movement

Along with the AFP this plane provides the other main component of the primary view. Unlike the static nature of the AFP this is a dynamic view showing all tracked movement within the zone, through live data feed from the tags, and offers various interactive options for the operator.

Selection of a coloured dot – used to denote individuals within the prison – displays details on a side panel or banner for that individual. While the colour of the dot is intended to identify the classification of the individual (staff, guard, inmate, visitor, etc.), it was also considered that these would be defined by the security category of the inmate. As this could begin to require operators to learn a prism of colour codes the “data click facility” allows them to get details immediately on the screen rather than remembering colour meanings. It is envisaged that one form of colour-coding would be maintained to make the initial view as valuable as possible.

To further enhance the information available at first glance of the screen the system will have a “multi-banner-click” facility. In situations where the operator may wish to keep track of several individuals they may click on several dots to generate a banner attached to the dot that gives the inmate or staff number with a base colour indicative of the inmates risk category.

This is an alternative configuration to the original notion of a checkbox next to the inmate’s details that highlights their dot on the screen. This is because in larger situations it may not be feasible to show multiple subjects’ details on the screen.

An alternative, though more user intensive solution, would be the option to multi-select as before, but then have the facility to request a new view that specifically matches the selected individuals. An algorithm can be used to define a radius around each dot (in a similar way to how the halos are calculated using a predefined radius and the co-ordinated grid). From this the system can determine the necessary floor plan view to watch all selected targets at the greatest resolution.

3. Tracking System Interface

The previous section looked at the pre-existing software planes and their use within the operator interface as well as some functionality they may offer. We will now look in more detail as to how this interface can be achieved with some additional functionality.

There are a number of elements that the operator needs to see:

- Movement of individuals within a designated zone
- Movement between zones / access points
- On demand details of individuals
- Details of individuals involved in collisions
- Details of multiple individuals selected by operator
- Exact location and category of collisions
- Tardy Response Mechanism (TRM)[1] status
- Request for intervention for TRM
- Individual’s records
- Collision specific view
- Monitored zone
- Master zone
- Sweeping and zooming capabilities
- Tag health concerns

The following sub-sections will cover how the above may be achieved.

3.1 Representing the floor plan

As mentioned in the previous section there is an inevitable balancing act in any software design between getting the most out of the GUI while not requiring a mainframe to get the system to run at a respectable pace.

With this in mind the floor plans must be represented in such as way as to be instantly recognisable to the operator, using logical objects to represent common features such as doors, windows, gates, etc. but keeping the graphical density to a minimum so that performance is not inhibited when panning, tilting (for three-dimensional systems), sweeping or zooming for instance.

Representing these zones as individual views is fairly straightforward where the prison architecture remains basic with wall-enclosed floors, walkways and stairwells. However, many prisons utilise a more open-plan, highly visible layout that makes use of open stairwells and mezzanine walkways, often opening onto large areas such as recreation halls [2]. Design decisions must include an ability to view these multilayered zones in a logical way that allows the operator to isolate problem spots (as previously considered in section 2.5), or view the whole zone with clear appreciation of the vertical orientation of the mezzanine floor observed. Such complex designs may require a three-dimensional view, that allows the operator to pan 360°. Alternatively, the operator has two views of the zone: birds eye and skyline, the latter denoting the focal point of the former.

Floor plan views will be divided up into predefined zones for ease of monitoring. Within each monitored zone the operator can zoom in and out (e.g. so that their view encompasses more than one zone if desired), and sweep to enable them to move their view around the master floor plan e.g. if they wished to follow activity to another zone. This is particularly important for tracking movement between zones as this is largely managed by the system itself [7] and so will be of special interest to operators who no longer need to manually approve access. A master view should be available to allow operators to drill down to selected zones or as a primary view that highlights collisions or triggered TRMs identifying areas of interest for the operator.

3.2 Operator Views

The operator interface must include a number of elements to allow the operators to effectively monitor activity within the prison. In addition to the floor plan, they must also be able to track movement of all individuals, call up individual data at speed, identify and tag individuals for special attention and be kept abreast of triggered Tardy Response Mechanisms and their status – most importantly when intervention is required (see section 3.3 User Interaction).

For the system to optimise its value and that of its autonomic components it is imperative that operators are immediately made aware of any halo collisions, access violations (or attempts there of), and tardy responses.

All of these can be dealt with initially by the system with no operator intervention but they must still be aware that they have occurred so that they are ready to act if necessary or simply to make a note of them for trend analysis and problem prediction.

Subsequently the operator interface should ideally have a data feed panel across the bottom, below the floor plan view that details all such infractions. Required details would at minimum be:

- Inmate / staff number
- Full Name
- Risk category

In addition to the above infractions this panel should also offer details of any logical violations where the tag's movement has violated the logical rules governing their movements around boundaries.

3.3 User Interaction

The operator must be able to interact with the interface in order to request further information or change the view (as mentioned above).

Ideally whenever an individual was involved in a halo collision, tardy response or possibly moving through an access point their details would automatically be shown on the screen in a panel specific to the activity. This would require designated panels on the screen that update automatically when any of this conditions are met. Thus keeping the operator fully aware of exactly who was participating in the activity.

In addition, the operator should be able to select an individual or multiple individuals whom they wish to either track on the screen, or see additional details for. Such information cannot be displayed for all individuals due to inevitable performance overheads and cluttering of the screen. A simple solution would be a banner. A single click to any "dot" will display a small colour-coded banner with the individuals ID number and surname. The colour should denote both their status (staff/inmate/visitor) and in the case of inmates their risk category [1]. A second click removes the banner. The operator may select multiple "dots" as required.

The system as a whole must integrate with the existing CCTV surveillance system. Each floor plan view should include objects signifying security cameras. Clicking on a particular camera should load the live video feed from that camera. Such integration may add an unacceptable integration complexity. An alternative solution would be identification of each camera that allows the operator to know the video screen to check that corresponds to the camera object on the screen.

4 Enhancing the Autonomic Capabilities

The original design roadmap made use of several autonomic system constructs that allowed the system to function with minimal user input. These include:

- Self-* properties [5]
- Apoptosis (self-destruct) [3]
- Quiescent (self-sleep) [6]
- ALice [5]
- Lub-Dub [4]

This functionality was intended to reduce the staff overhead required to manage movement throughout the prison. Here we consider some further features that extend upon these initial ideas when considering what operators may wish to be made aware of and instances where human intervention is required.

There was also the consideration at the start of the paper of performance vs. hardware costs. Self-learning and self –optimising capabilities can be used here to enhance performance of the system at critical times or escalated behaviour only, thus continuing to optimise overall system performance.

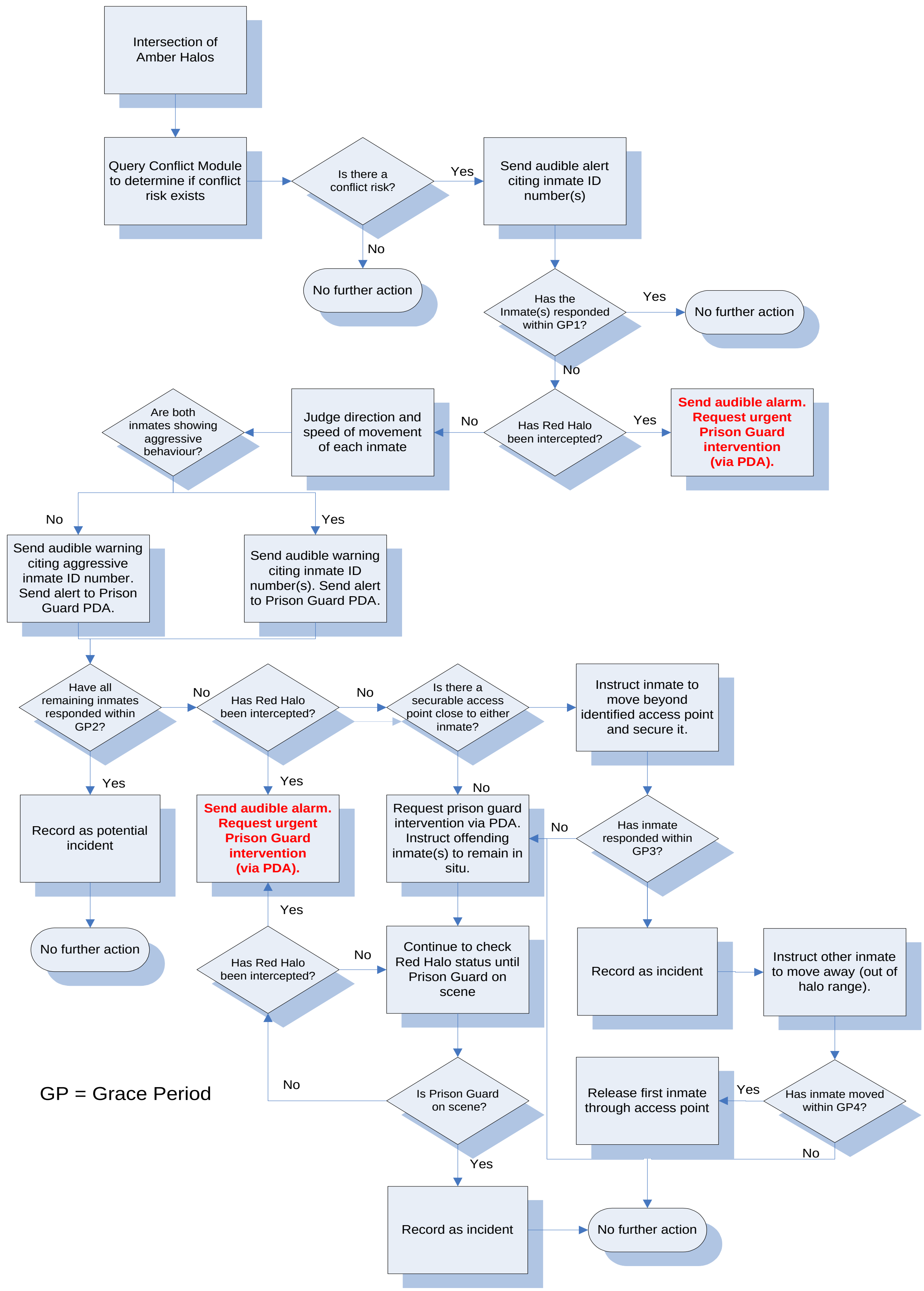

Figure 2. Tardy Response Mechanism (TRM)

4.1 Collision Highlights and Data Feeds

The most powerful element of the tagging and tracking system is its ability to handle access and movement violations (identified through halo collisions [1]) without human interaction via the Tardy Response Mechanism (Figure 2). There is still value in the operator being made aware of such violations and the interface must offer some capability to do this.

Initially these should be identified on the main screen within the floor plan view. This is the main focal point of the operator and so will grab their attention; it also allows immediate identification of the exact location of the violations. Flashing/highlighted objects – either the individual(s)' dot or the access point – should be used to draw attention to the areas.

As mentioned dedicated panels will report details of the involved individuals (3.2), however, as the system will handle the initial infraction control the operator needs a way of knowing what the system is doing and how far through the TRM process it is. Data feeds offer an easy solution for this. Ticker-tape feeds in a dedicated panel across the bottom of the screen can detail the stage the process is at as well as the individuals involved. Regular updates are posted as long as the process is live. This should occur automatically.

In instances where more than one violation in a given zone is occurring simultaneously a one-click option performed over the highlighted object should change the panel display to the selected infraction.

As human intervention is not required at this stage in the TRM process anything used to draw attention to an infraction should be fairly relaxed. As you will see in the next section, informational notification should not be able to be confused with actionable notification.

4.2 User Assistance Requests

As can be seen from Figure 2, the TRM process may reach a point where human intervention is required. This intervention request is sent to the PDA of the nearest prison guard in situ via the Control Panel Server [1]**.**To ensure the prison guard responds and possibly to offer backup the operator must be informed when the TRM mechanism has reached this critical "red" stage.

The interface requires a manner of notification that stands out from any previously suggested. To draw attention to the data feed panel, the text should change to red when this step in the process has been reached accompanied by an audible alert to draw the operator's attention to the screen.

4.3 Halo Recalculations

In 3.2 we considered that the operator should be informed of all infractions whether their input was required or not so that behavioural patterns may be learned and possible problems predicted and thus avoided in the future.

Infractions are identified by so-called halo collisions. A halo is a predefined circular area of co-ordinates extending around each individual. Each individual has an amber and a red halo whose co-ordinate range is determined from a prison specific defined radius (see figure 3). Whenever one of these halos either interacts with another halo or an access point the Tracking Module is alerted and the TRM is triggered as shown in Figure 2.

Autonomic computing focuses on so called self-* properties [5]. These enable a system to monitor its environment (self-monitoring) and adjust its processes according to that input (self-adjustment) for optimal performance (self-optimising).

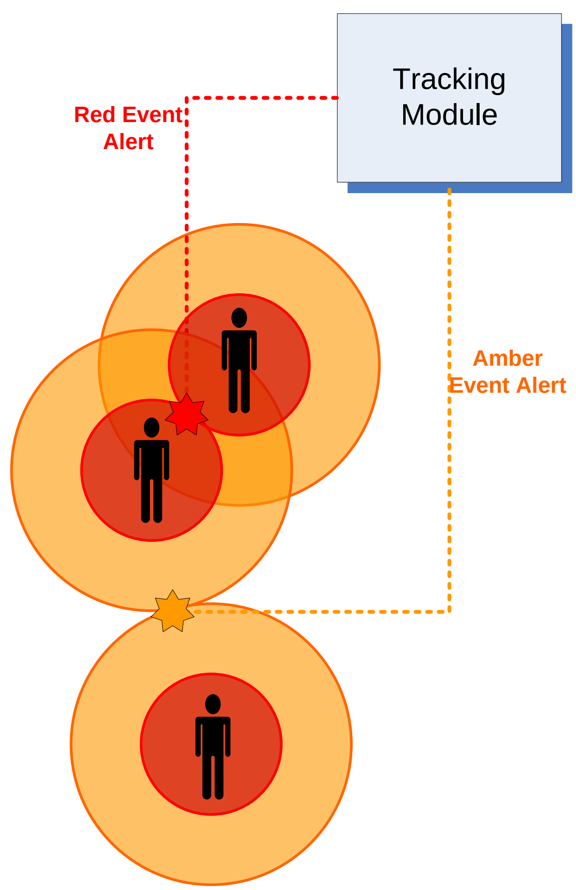

Figure 3. Halo collisions

These properties can be used in halo recalculation, enabling the system to adjust the halo radius based on feedback from halo collision and TRM data. As an example, if it is found that a particular inmate constantly violates the TRM as a result of a halo collision and goes on to assault other prisoners, the system can evaluate this data and readjust the radii, making them larger to ensure the process begins earlier and thus gives prison guards and / or operators more time to intervene before an assault takes place.

The recalculation will be based on such factors as:

- Frequency of TRM violations in a given time period against the same or different objects, e.g. the same access point
- Time (sec) from initiation of TRM and request sent to prison guard PDA – the smaller the time period greater the radii increase
- Correlation between initiated TRM and recorded incident - how often does the triggered TRM for this inmate result in an incident?

4.4 Tag feedback frequency correction

Continuing to focus on the self-* properties we can consider further how autonomic computing capabilities can help with the system performance.

It was previously determined that optimum tag feedback to the system would be at one second intervals [1]. However, it may be found that either the existing or available infrastructure cannot support this resultant volume of data feed or that there may be scenarios in which the data feed frequency needs to be higher or possibly lower than the default. This would be based on inmate activity.

Altering the feedback frequency allows the system to use less processing resources during quiet periods and / or for low risk inmates with a low violation probability. The latter in particular can then free up processing resources for high risk, high violation inmates as their tag feedback frequency is increased.

By utilising the self-awareness of an autonomic system the system itself can evaluate what is happening in the environment and adjust its own settings offering optimum system performance as regards both the speed the application performs at e.g. screen refresh rates and input response times, and in gathering and disseminating data about suspicious activity at a higher than normal rate to offer more granular monitoring, e.g. when the TRM is activated.

5. Conclusion

Although the ultimate vision of Autonomic Systems is to remove the human from the management loop it is still critical for humans to have effective interfaces to the self-managing systems especially when human’s exhibiting autonomous (and potentially non-desirable) behaviour are actors within that system.

This paper considered an interface design for such a system with contributions on conflicts, halo collisions and tardy response mechanisms tracking movements of inmates in correctional facility, based on the next generation tagging and tracking roadmap [1], for ‘management’ in the loop– the prison officers.

Acknowledgements

The wider context of the Autonomic Systems and Smart Environments research is supported at the University of Ulster by the Computer Science Research Institute (CSRI).

References

[1] Mulholland CL, Sterritt R, OHagan P, Hanna E, (Mar 2008) "Tagging and Tracking System for Prisons and Correctional Facilities – A Design Roadmap", Proceedings of Fifth IEEE International Workshop on Engineering of Autonomic and Autonomous Systems (ease 2008), Belfast, Northern Ireland, 31st March - 4th April, IEEE CS Press, Pages 143-153

[2] Fairweather L., McConville S., “*Prison Architecture*”, Architectural Press, 2000.

[3] Sterritt R, Hinchey MG, "Apoptosis and Self-Destruct: A Contribution to Autonomic Agents?", Proceedings of Third NASA-Goddard/IEEE Workshop on Formal Approaches to Agent-Based Systems (FAABS III), Washington DC, April 26-27, 2004, in "LNAI 3228", Springer-Verlag, Dec 2004, Pages 262-270

[4] Sterritt R, Hinchey MG, (Jun 2005) "SPAACE :: Self- Properties for an Autonomous & Autonomic Computing Environment", Proceedings of Autonomic & Autonomous Space Exploration Systems (A&A-SES-1) at 2005 International Conference on Software Engineering Research and Practice (SERP'05), Las Vegas, June 27-30, CSREA Press, Pages 3-8

[5] Sterritt R, Hinchey MG, *"Radical Concepts for Self-Managing Ubiquitous and Pervasive Computing Environments*", In proceedings WRAC-II, 2nd NASA/IEEE Workshop on Radical Agent Concepts, Sept. 2005, in LNAI 3825, 2007. DOI: 10.1007/11964995_33
[6] Sterritt R, Hinchey MG, "Biologically-Inspired Concepts for Autonomic Self-Protection in Multiagent Systems", Proceedings of 3rd International Workshop on Safety and Security in Multi-Agent Systems (SASEMAS 2006) at AAMAS 2006, Hakodate, Japan, May 2006
[7] Sterritt R., Garrity G, Hanna E., O'Hagan P., *"Autonomic Agents for Survivable Security Systems"*, Proceedings of 1st IFIP Workshop on Trusted and Autonomic Ubiquitous and Embedded Systems (TAUES 2005) at EUC'05, in "LNCS 3823", Springer, December 6-9 2005.
[8] O'Hagan, Patricia, Hanna, Edward and Sterritt, Roy (Feb. 2010) "*Addressing the Corrections Crisis with Software Technology*". Computer, 43 (2). pp. 90-93. ISSN 0018-9162, DOI: 10.1109/MC.2010.29