# AN AGENT-BASED FRAMEWORK TO ENABLE INTELLIGENT GEOCODING SERVICES

**M. J. Hutchinson, B. Veenendaal**

Dept. of Spatial Sciences, Curtin University of Technology, Perth, WA, Australia - b.veenendaal@curtin.edu.au
Cooperative Research Centre for Spatial Information, 723 Swanston St, Carlton, VIC, Australia - mj.hutch@gmail.com

**Commission IV, WG IV/5**

**KEY WORDS:** Internet/Web, GIS, Learning, Abstraction, Artificial Intelligence, Georeferencing, Knowledge Base

**ABSTRACT:**

Geocoding is essential to translating a physical address such as a house, business or landmark into spatial coordinates. These coordinates represent location which is an essential ingredient for location based services and web mapping. Despite progress in the field of geocoding, there remains a sizable proportion of addresses that are difficult to geocode. The purpose of this research was to explore how agent-based processing, which utilizes the belief, desire, intention (BDI) model, can add intelligence to the geocoding process. The event driven nature of agent-based processing makes it suitable for use in a web service platform. The event-driven and reactive behaviour is complemented by the ability for goal directed and non-deterministic behaviour in the longer term. Overall control of the geocoding process is based on the interactions between agents which represent the geographic elements contained in an address. Each of these agents operate in parallel, pursuing tasks associated with correcting and preparing their individual element for geocoding. This results in a geocoding process that has multiple foci of control and is iterative. The same geographic relationships that exist between the address elements also exist between the agents. Messages are used for the agents to send and receive data between themselves, and because the agents represent geographic elements, these messages have content relating to their real world geographic relationships. These relationships form the basis of the inherent semantics in the intelligent framework. Results indicate that intelligence in geocoding is a product of both context and semantics, at a conceptual level, and control and knowledge, at an implementation level, where the two are "connected" by the agent paradigm which is both a representation and a solution. This paper presents the agent component of the framework used to enable intelligent geocoding and the results from a prototype implemented using the intelligent framework.

## 1 INTRODUCTION

This paper outlines the use of agents in geocoding. Note that a rule-based system was additionally used to create a knowledge base and provide an inference capability for learning in geocoding, however this aspect of the research is beyond the scope of this paper; subsequent papers will describe the rule-based system used by the agents.

### 1.1 Current and Future Geocoding

The geocoding process is well documented in the literature (Goldberg et al., 2007; Karimi and Durcik, 2004; Zandbergen, 2008; Wei et al., 2009; Lee, 2009; Jacquez and Rommel, 2009). The geocoding process relies on a combination of techniques from record linkage and GIS. The record linkage is necessary to take the original address submitted for geocoding and to accurately identify corresponding records in one or more reference data sources which describe that address (Gu et al., 2003) for the purposes of correcting and verifying the address. After linkage occurs, techniques from GIS assist in assigning geographic coordinates to the address. Geocoding is at a transition point where its current capabilities are going to be confronted by advances in the Internet, artificial intelligence and geocomputation. Society is on the verge of a semantic geospatial web (Egenhofer, 2002), and this mindset needs to be embraced for the field of geocoding. Geocoding must follow the semantic geospatial web as it brings a totally new way of organizing information (Egenhofer, 2002; Passin, 2004) with the ability to obtain spatial location based on the meaning of spatial and textual queries.

It has been suggested by Berners-Lee (2001) that when software agents are collecting web content from diverse sources and processing the information and exchanging the results with other agents, that the real power of the semantic web will be realized. Berners-Lee (2001) has also suggested that the effectiveness of these programs will increase exponentially as more machine readable web content and automated services become available. This has implications for both the creation and use of information, along with the control of how this information is used in software. With the opportunity afforded by these emerging technologies, the question is how will these affect the phases in the contemporary geocoding process and benefit the current limitations and difficulties in contemporary geocoding? Geocoding has an established linear work flow with certain steps, and these should not be thrown out; but the way they are executed and controlled could be improved. Renovating the geocoding process with a new mechanism for control could not only prepare it for the burgeoning semantic geospatial web, but would also provide the opportunity for greater extensibility and modularity.

### 1.2 Agent-Based Paradigm

Padgham and Winikoff (2004) define an agent as being "a computer system that is situated in some environment, and that is capable of autonomous action in this environment in order to meet its design objectives". Because agents are situated in an environment, they are instantiated once and continue to run in memory until stopped.

In addition to this, Wooldridge (2002) defines an intelligent agent as also being reactive, proactive or social. Even when nothing is happening (i.e. the agent has an empty agenda) the agent is still instantiated however it sits idle waiting for new events. Every catalyst for the agent doing something is an event of some form (either internal or external). Some applications use only one agent, but often more than one agent is required due to complexity or conceptual modelling; these are called multi-agent systems

(MAS) (Ferber, 1999). MAS are also referred to categorically as distributed artificial intelligence systems (Jennings, 1993); the defining concept is that "multiple agents interact to improve their individual performance and to enhance the system's overall utility" (Jennings, 1993).

Interaction and autonomous behaviour in particular are significant capabilities with regard to multi-agent systems (Jennings et al., 1998). Agents are capable of sending messages to each other, which can be as simple as a single message (to which a response may not even be sent) or as complex as a whole session of messaging, using established protocols between one or more agents. The autonomy means that within a MAS each agent is pursuing its own goals without regard to what the agents are doing. It is up to the individual agents whether or not they initiate messaging and respond to requests. Depending on how the system is designed, the MAS may have competitive or cooperative agents. Each has their own uses, although a cooperative system is very useful as tasks can be achieved in parallel. A specific architecture within goal-based agents is the belief, desire, intention (BDI) model. Desires can be thought of as goals the agent wants to achieve, intentions as plans that dictate how to go about this and beliefs are internal data (Agent Oriented Software, 2003). The BDI system is modelled on ideas from psychology and philosophy, simplified into a version suitable for computer implementation (Howden et al., 2001). Just as people have a view of the world, the BDI model sees agents also having a "view" of their world; this view is represented by the beliefs in the beliefset (Kinny et al., 1996).

## 2 AGENT-BASED GEOCODING

To demonstrate this research and the intelligent geocoding framework, a prototype has been generated using MAS techniques. One of the main aspects of the design is that overall control of the geocoding process is based on the interactions between agents representing the geographic elements contained in an address. In other words, each address element is represented by a software agent and each of these agents pursue tasks associated with correcting and preparing their individual element for geocoding. Because there are multiple agents, these tasks occur in parallel. The result is the geocoding process in software running with multiple foci of control. As each agent is correcting its own element, all the elements from each of the agents are reassembled and coordinates are found for the address (this is ongoing and occurring in real-time).

### 2.1 Agents as Geographic Elements

The agent types assigned within the system resulting from this research include a user agent, matching agent, and five specialty agents. The specialty agents are named the state, postcode, locality, street and property agents. Note that the street agent is responsible for processing both street name and street type. The property agent is responsible for processing street number and unit number. When the system is run, multiple instantiations of this design are created and each assigned to a particular role.

Because agents are used to represent the geographic address elements, the same geographic relationships that exist between the address elements also exist between the agents. However, because the agents represent geographic elements, these messages have content relating to their real geographic relationships. Examples include, the relationship that a state contains a particular locality, or that a street is within a given locality. Although five agents are mentioned specifically, the design is expandable to cater for more. The geographic representation does not have to stop at the property level, and the design would allow drilling down to buildings, rooms in buildings and even objects in rooms. This adds flexibility and extensibility to the design, and is not constrained by the increasing complexity that would happen in a "linear" environment. Likewise, in terms of the higher levels related to areal features, it does not have to stop at the state. Beyond state, there could also be a country representation, this in particular would work well with semantic knowledge detailing the idiosyncrasies of geocoding in other countries (e.g. geography, topology, temporal). The geocoding process is the same, with a few differences, for each of the address elements and this lends itself to reusing large parts of agent behaviour for the various agents.

### 2.2 Agents as an Abstraction and Implementation

It is because the agent is the geographic element, and vice versa, that a commonality exists between control, knowledge and the phenomena being abstracted (the geographic element). This commonality is such that the geographic elements also benefit from receiving context and semantics. Figure 1 presents how the intelligent framework can be thought to have two imaginary tiers involved, where the bottom tier is the "behind the scenes" use of control and knowledge which in turn drives the context and semantics on the surface.

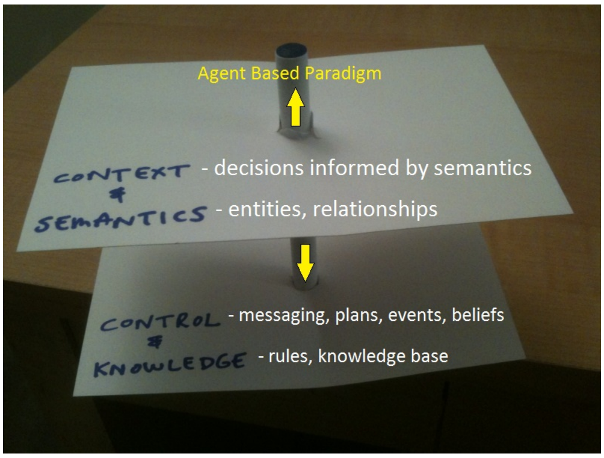


Figure 1: Agent based paradigm as a bridge

Also, because the two "layers" of intelligent geocoding are parallel and aligned, then conceptually any effect or capability on the top layer has a corresponding cause or capability on the lower layer. The vertical column in Figure 1 represents that it is the agent paradigm which ties the two layers together and allows for the duality of the control and geographic representation.

## 3 RELATIONSHIPS BETWEEN AGENTS

### 3.1 Message Communication

Because the processing of individual elements does require information about other elements in the address, messages are used for the agents to send and receive data.

### 3.2 Inherent Semantics

When working with the geographic elements, they are arranged conceptually as seen in Figure 1, where they are arranged in order of which elements "contain" other elements. This arrangement is what underpins the semantic relationships between the agents representing the address elements.


Figure 2: Geographic relationships of address elements

There are several terms that describe the relationships between elements. An element is present if a value was submitted in the original submitted address, i.e. the element value is not blank. An element exists if it can be found in at least one reference data set, e.g. the locality "Wembley" exists in the gazetteer for Western Australia.

As seen in Figure 1, a given element has other elements on the "left" and "right" of it; these are referred to as *complementary elements*. For example, the complementary elements for the postcode are the state and locality. In a real query, if a particular element type is not submitted (i.e. it is blank) then the complement is the next present element type. The state and unit number elements do not have upper and lower complements, respectively. The complementary elements in this research are address elements, but there is no reason they could not be more generally used as "complementary components", any component that better contextualizes it and gives additional assurance. This agreement of complements is a spatial agreement, for example the postcode contains a given street, or a street does have a particular street number on it. It is possible that two elements may not agree spatially, even if they both exist; this case would mean that one of the elements is incorrect.

An *equivalent complement* is the equivalent, complementary element value for a given element value; the result is selected from a reference dataset, using the given element value as the search criteria. For example, using a street name, its equivalent postcode can be found; local datasets can be used for this, or web services which provide a rich choice of data from a variety of providers with the benefit that data does not have to be maintained locally. For the research, web services provided by LISAsoft. Using this approach, an element type finds the equivalent complementary element only to its "left" (the containing geographic element); this is an effort to perform a "one to one" search as often as possible (e.g. a street usually has few or a single locality associated with it, but a single locality would have many streets in it). This "one to one" idea is used simply to keep the list of equivalents as small as possible.

## 4 PROPERTIES OF AGENT-BASED GEOCODING

### 4.1 Parallel Behaviour

A user agent was created to manage the brokering of queries from users, and then distributing these to the element agents. This user agent provides a neat start and finishing point for the parallel processing done by the element agents; it is the user agent which has the coordinating role for the multiple queries coming in. It is within the user agent that each agent is assigned a unique ID and kept track of so that the outcomes can be distinguished. The user agent is responsible for "distributing" these identifiers as needed by the agents, and when closing a query it retires the identifier. Agents can be used in parallel within a single machine (multiple cores) or with several machines (distributed processing). It should be noted that for every additional query coming into the system, no additional agents are being created. The same agents are instantiated the whole time, and remain instantiated between queries so there is no overhead in creating new agents. Memory is used to process events, send messages, and add beliefs to beliefsets (and other operations), but it seems logical that this processing would scale as additional agents of different types could be added. For example, there is no reason there could not be five locality agents (or five of every agent) and when new geocode queries enter the system they could be assigned appropriately (load balanced) to an agent.

### 4.2 Goal-Based and Non-Deterministic Processing

The goals and sub-goals within the framework can be seen in Figure 3, where a goal is represented with a box, sub-goal as a coloured ellipse and a plan as plain ellipse.

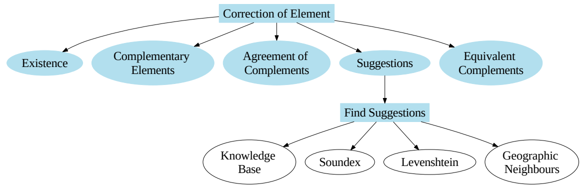


Figure 3: Goals and Sub-Goals within the Framework

The goal "Correction of Element" is the top-level goal which is pursued for every address element (i.e. every agent pursues its own instantiation of this goal), and in turn each of the sub-goals are also pursued. Each of these sub-goals is a step in the element correction process, although depending on the status of the element, not all steps may be required at a given time in the geocoding process. Looking at these sub-goals, and their role in the geocoding algorithm, it is seen that (i) goals are applied in a sequence where order is important, (ii) not all sub-goals will be needed in every situation, (iii) goals are subtly influenced by their environment, and (iv) a mechanism is needed to ensure one goal is completed satisfactorily before moving on to the next. The goal to test whether an element is present and exists ("Existence") is a simple lookup; no techniques are used in this step to try and find a possible replacement or correction. This information is stored in beliefsets. There are some special considerations to this step, such as when a particular type of element is required to check the existence of another because the other by itself is simply not possible; e.g. to check if a street number "exists" at the very minimum a street name is also needed. The "Complimentary Elements" goals sends messages to other agents (which each represent elements) to ask what their values are, which determines what the complimentary elements are for the original element; these values are then stored in the beliefset of the original agent. Each agent exchanges messages with at least one or two other agents to do this, and this is an example of activity occurring in parallel. It is important to mention that related to the use of these goals (and overall geocoder control) is the role of address element status scores. These scores are calculated for each address element and also for the overall address. The score of an individual agent affects messages sent out to other agents. When certain beliefs are true then the agent is "eligible" to do perform certain actions and interactions. For example, once an agent has determined whether its address element is present and exists, it will respond to other agents requesting what its status is. Previous to determining whether its own element is present and exists, the agent will not respond to enquiries from other agents, ensuring it has basic information regarding its own status. If an agent reaches an individual score of 1.0 then this is the catalyst for the agent indicating they are "complete", and eventually when all agents are complete then the processing of the overall address is complete. Similarly, even when agents reach a final maximum score less than 1.0, this is communicated via messaging and the agents reach consensus indicating that overall processing is complete.

### 4.3 Distributed Scoring

As seen in Section 4.2, a key component to agent behaviour is the scoring algorithm used in the prototype, which is both distributed (across the various agents) and weighted. The focus of quality in the agent system is on the quality of the address results, specifically to what extent the address elements agree with each other and the overall spatial agreement of the address. Following from this, there is a quality score calculated for each of the individual agents and an overall score the whole address.

The quality score for an individual element in the address has a value between zero and one. The score is a measure of how well an element "fits" with the address it is in. A value of zero (0) indicates no agreement, while one (1.0) indicates that the element agrees completely with its neighbouring elements. Combining these individual scores together can also provide an overall measure of quality for the address.

The individual score is calculated by using several factors, including whether the element value is present, exists, and agrees with both its upper and lower neighbours (where applicable). Each of these factors is also given a numerical weighting, which reflects the relative importance of the factor. For example, when calculating the score for street name, it would be considered much more important for the street name to agree with the locality than with the street type; this is reflected by giving locality a larger weighting in the calculation. Each particular element being calculated can have different weightings for the same element type. Also taken into account is the current score of the element's complementary elements. This means that the current status score of one element will affect the score of the elements that rely on it, because of this the element scoring process is interdependent and gives an inherent measure of the overall address quality. The calculation be seen in Equation 1, where *w* is the weight and *s* is the score.

$$\text{score} = \frac{w_{present} + w_{exist} + (w_{upper} \times s_{upper}) + (w_{lower} \times s_{lower})}{w_{present} + w_{exist} + w_{upper} + w_{lower}} \quad (1)$$

When calculating the numerator, the weight for present and exist are only added if the element is present and exists. When calculating the denominator, the weights for being present, existing and agreeing with the neighbours is added regardless. Ultimately, this means that the status score for an element is penalised via the numerator. When a new query is submitted, each element is given an initial quality of 1.0. This can be thought of as providing the element with the "benefit of the doubt" regarding its score; not until proven otherwise is the element penalised. A score for the overall address can be calculated by combining these individual address elements. This formula can be seen in Equation 2, where n is the number of elements submitted in the original query, and m is the number of elements in the final, matched address result for which a geocode is returned to the user; when calculating the overall score before processing has completely finished, m is the number of elements which are present, exist and spatially agree with their complementary elements. The score of the element is the same score calculated for individual elements in Equation 1. It can be seen in Equation 2 that the score is penalised if the number of elements in the final matched address is less than the number originally submitted. This penalty is applied because the resulting address contains less information than the original. This penalty is useful to quantify the fact that although a result address may have a better score than the original submitted address, its geocode may have a reduced resolution than the original query.

$$\text{score} = \frac{m}{n} \times \frac{\sum_{i=1}^{m} (w_{\text{element } i} \times s_{\text{element } i})}{\sum_{i=1}^{n} w_{\text{element } i}} \quad (2)$$

Each element used in Equation 2 has a weighting which denotes its importance to the overall address. For example, the postcode can be configured to be more important in determining a geocode than a street type.

As well as providing the user with an indication of how reliable their results are, the quality measures are also important in internal system processing. When multiple address matches are found, regardless of whether one or many results are expected, these matches need to be sorted. There are several mechanisms for sorting, but one of these is the whole-address quality. For example, although there could be two matched addresses with same number of correct elements, because of the weightings associated with the different element types, this means that one of the addresses could have a higher whole-address score calculated for it and subsequently considered a better result.

### 4.4 Iteration during Correction

There is a concept of "parent" and "child" addresses, and more specifically, child elements. The original address is considered to be the initial parent, or root node, of the system. During geocoding, several iterations can occur depending on how incorrect the original address is. Suggestions are made for each element that does not exist, or exists but does not spatially agree with its complements; each of these suggestions adds to the number of possible addresses that could be the "corrected" version of the original query. The iteration within the system stops after three child "branches" have been pursued, or when a complete match is found. If extending the iteration beyond three branches, there may be a risk of straying too far from the original query element from both a semantic and practical perspective. Each suggestion that emerges from the matching and is kept becomes a child node. Each element within the original address is essentially the parent node of its own tree; and the parent node along with each of the child nodes have corresponding, complementary elements on the other trees.

Figure 4 shows that each element suggestion has three unique integers which together distinctly identify the element. These three integers, from left to right, are the *query-ID*, *sub-ID* and *parent-ID*. The query-ID is the same for every element contained/generated from the same original query submitted by the user; in Figure 4 this value is 1. The sub-ID is unique for every element, this simply increments for every additional element suggestion that is added; in Figure 4 this value ranges from 0 (in the root element) to 6 (the final element or third generation).

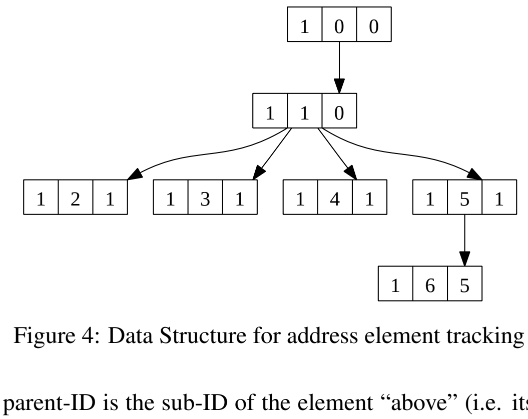


Figure 4: Data Structure for address element tracking

The parent-ID is the sub-ID of the element "above" (i.e. its parent) any given element. It should be noted that all of this processing is stored in beliefsets, and this is just one example of how beliefsets are useful as mechanism for dynamic storage, querying and triggering.

### 4.5 Multiple Foci of Control and Recursion

The "Correction of Element" (implemented as *FindBestValue*) goal seen in Section 4.2, has several criteria for success, one of these criteria utilizes the concept of parent and child nodes seen in Section 4.4. The goal succeeds when a parent element has more than zero child elements, and the number of total child elements equals the number of completed child elements. As a design decision, no more than three "generations" of child element suggestions are allowed to ensure the suggestions are not too far removed from the intended value inputted by the user. This means that processing would stop when an element writes to a beliefset a parentID of 2. Of the different criteria for termination, it is the first to occur which causes termination. In Aldemir (1994), Friedmann Mattern's idea of "sticky state indicators" is provided as a solution for this distributed termination detection problem which useful for termination in agent-based geocoding.

This tree structure of nodes and the FindBestValue is recursive, in two ways. The first is that suggestions originate from suggestions, and this continues for three generations. As a result of this, the second form of recursion is that the goal FindBestValue will be nested. This is because a parent element will not have completed its goal until its child element has, and so on; once the child element has finished, the parent is free to complete their goal. The steps after the posting of FindBestValue are evaluated every time the FindBestValue is posted, although not every sub-goal will necessarily be pursued. This capability for tracking address elements throughout the system, and knowing which elements are the parents or children makes possible the concept of iteration. Iterations allows the cycle of goals to be used several times. Each of the goals are visited one after another, which at first inspection seems to be the same as the linear geocoding process used in current geocoders, but the whole sequence of goals (or a selection) can repeated if needed. The rationale behind this is that some addresses may need several transformations in order to be fully corrected. For example, a user may have a locality confused with its neighbour and may also spell that neighbour incorrectly. It is expected that in the future that the use of in-depth semantics in geocoding will further validate this concept. If the whole geocoding process can be thought of a "pipe", the iteration essentially takes the first set of results and drops these back into the pipe; this is how the current linearity of geocoding is combined with the distributed, goal-based and iterative framework to form a hybrid solution.

## 5 RESULTS

The results presented in this paper are in terms of (i) the behaviour demonstrated by the intelligent geocoder prototype, and (ii) the qualitative benefits from derived functionality made possible by the agent-based approach.

### 5.1 Intelligent Geocoder Behaviour

The scoring mechanism which was distributed and weighted catered for the distributed and interwoven geographic relationships of the prototype, and provided a meaningful metric which underpinned much of the event driven messaging and behaviour.

One of the most interesting results was that scoring reached a natural maximum based on consensus (i.e. agents messaging updates between themselves), rather than a single overall control mechanism telling the agents to stop at a pre-determined score. The messaging system was designed such that although agents could update each other with score increases (and subsequently update other agents), there was not a frenzy (a race condition) where scoring continued incorrectly and continuously. An example of the locality element reaching a natural maximum score of 0.71 can be seen in Figure 5.

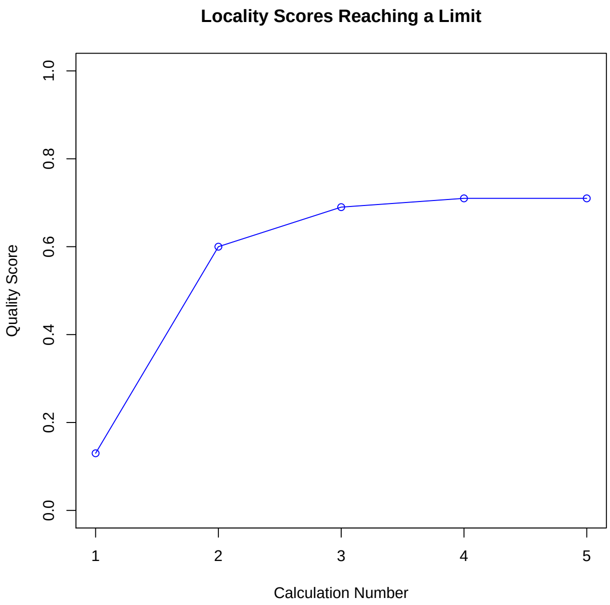


Figure 5: Locality Element Score Change Reaching a Limit

It was suggested in Section 2.2 that agents were both an abstraction and an implementation; the scoring mechanism used for the prototype fits with this approach, as it bases scores on the abstraction (geographic elements) but is made possible by the implementation (agent messaging, beliefs, goals). Having weightings for individual elements, scores for individual elements and an overall address score make for clear user awareness and quality communication. Including a measure of resolution was also intended to better communicate the final address quality.

Another interesting observation was how the activity load (number of calculations needed to reach a final score) varied for each agent type. It was found that activity load for a given element (e.g. locality) was affected by the number of other address elements that provide updates to that given element, and the number of increments that other elements have will affect the number of increments for the given element. An example of the activity load (for an address in which all elements eventually obtain a score of 1.0) can be seen in Figure 6. This example shows how the locality and postcode elements have more calculations because of the number of other element they interact with and the number of messages they receive. The number of interactions (i.e. time) is also a function of the type of neighbouring elements it relies on and the weightings of all elements.

As seen in Figure 6, the difference in the number of increments for each element highlights how the algorithm (and in particular the ordering and time used for processing) is address-driven. During processing if an address is submitted in which one or more of the elements do not exist, then the agent still shows raw results for those elements that do exist. The idea here is that while the user is waiting for the correction process to correct the elements with errors, at least they can get preliminary geocode information (i.e. the user's display is updated in real-time). Parallel design has the advantage that thought only has to be put into each ele-

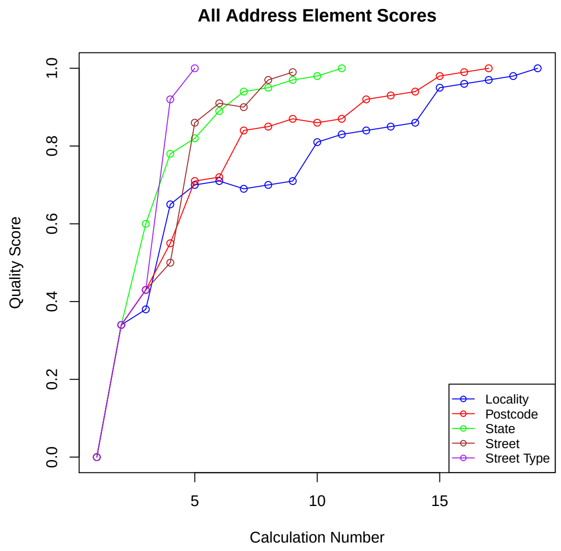


Figure 6: Address element score progressions

ment individually (i.e. control between agents does not have to be explicitly defined), as the resulting outcome emerges by itself.

### 5.2 Processing Addresses

Several categories of address types were used for testing, with each category containing a selection of representative addresses. In this paper, discussion of address testing is limited; subsequent research will contain additional detail regarding the address testing procedure, results and analysis. The testing was intended to (i) show that the agent-based framework is a legitimate option for geocoding, (ii) identify where the prototype provided advantage and areas where it could be improved, (iii) identify where correction techniques fit within the framework, and (iv) test the process of geocoding rather than assess the positional accuracy of the results, and (v) show that an agent-based framework does not have barriers with regards to adoption by industry. It is important to note that the prototype did not perform any standardization on incoming queries, and for retrieving geocodes other web services and pre-geocoded datasets were used; this reflects the fact that the framework is an intelligent mechanism for coordination within a web-based environment just as much as it is a geocoding engine. Not only does the prototype utilize web services, it can also deliver value-added services. A key point to remember is that it is the correction techniques, not the control framework, which are largely responsible for improving the errors in the geocoder queries themselves. The approach used for the intelligent framework is that standard (and emerging) correction techniques can be better applied by using an agent-based approach. The research also suggests new ideas for correction techniques, however the goal of the research was to establish a new framework for geocoding having real implications, rather than to solely write a collection of problem address correctors.

## 6 CONCLUSIONS

The most significant conclusions from the agent component of the research regard extensibility, the need for additional address correction techniques (for types of address errors not yet catered for) and the benefits of goal-based and non-deterministic processing. The outcomes of this research concluded that that agents provide a solid foundation (inherently containing many desirable properties) for the future of intelligent geocoding services. The framework has shown that geocoding using a collection of distributed elements/factors is feasible, and that these elements can be weighted individually and tied together for overall scoring. Adding additional factors (including virtual factors, i.e. not physical) in the future would be possible without having to re-architect the design because the factors can be represented as agents and more messages can be used along with weightings for the additional factors.

So how does this benefit the small percentage of difficult addresses and situations which continue to confront geocoding? It has been identified that there could be deeper causes for these address errors and issues (alluded to in the discussion of iterative processing), including errors based on semantic and spatial cognition. The key to this is that the intelligent framework is ready for these correction modules to be "plugged in", i.e. now specific correctors/modules can be written and incorporated into the intelligent framework which provides an environment suitable for considering semantics in addition to syntactic issues.

Looking at the overall power of the framework is enhanced by:

- Geographic relationships woven into the system design at a fundamental level (part of the paradigm)
- A shared design structure with the knowledge base (to be discussed in another paper)
- Inherent tracking, quality, scoring and context
- The intelligence provided by the agent paradigm

Intelligence within the agents comes from the ability to pursue goals over time, contextually select execution, and dynamically re-plan. The research in this paper has only begun to uncover and exploit the BDI approach to geocoding, much more is possible. The intelligent geocoding framework provides several advantages as a web service platform, it is inherently distributed, message based, has context and quality included in the framework at a fundamental level, and could use the goal-based processing and non-deterministic behaviour to query and combine other web services. At the core of the agent approach is an event-driven model which is suitable for a service oriented geocoding architecture. Agents can successfully serve as the bridge between context, semantics, control and knowledge by means of its paradigm. The intelligence derived from this provides advantages now and for the future.

## REFERENCES


Agent Oriented Software (2003). "Jack Agent Manual". http://www.agent-software.com

Aldemir, T. (1994). *Reliability and Safety Assessment of Dynamic Process Systems*. Berlin: Springer-Verlag.

Berners-Lee, T., J. Hendler, and O. Lassila (2001). The semantic web. *Scientific American* (284), 34 - 43.

Egenhofer, M. J. (2002). Toward the semantic geospatial web. In *Proceedings of the Tenth ACM International Symposium on Advances in Geographic Information Systems*, McLean, Virginia.

Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*. New York: Addison Wesley.

Goldberg, D., J. Wilson, and C. Knoblock (2007). From text to geographic coordinates: The current state of geocoding. *Journal of the Urban and Regional Information Systems Association* 19(1).

Gu, L., R. Baxter, D. Vickers, and C. Rainsford (2003). Record linkage: Current practice and future directions. Technical report, CSIRO Mathematical and Information Sciences.

Howden, N., R. Ronnquist, A. Hodgson, and A. Lucas (2001). Jack intelligent agents - summary of an agent infrastructure. In *5th International Conference on Autonomous Agents*, Montreal, Canada.

Jacquez, G. and R. Rommel (2009). Local indicators of geocoding accuracy (LIGA): theory and application. *International Journal of Health Geographics* 8(1), 60.

Jennings, N. R. (1993). Commitments and conventions: The foundation of coordination in multi-agent systems. *The Knowledge Engineering Review* 8(3), 223-250.

Jennings, N. R., K. Sycara, and M. Woolridge (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems* 1(1), 7-38.

Karimi, H. and M. Durcik (2004). Evaluation of uncertainties associated with geocoding techniques. *Computer-Aided Civil and Infrastructure Engineering* 19(3).

Kinny, D., M. Georgeff, and A. Rao (1996). A methodology and modelling technique for systems of BDI agents. In *Agents Breaking Away: Proceedings of the Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World* (MAAMAW96), pp. 56-71.

Lee, J. (2009, January). GIS-based geocoding methods for area-based addresses and 3D addresses in urban areas. *Environment and Planning B: Planning and Design* 36(1), 86-106.

Padgham, L. and M.Winikoff (2004). *Developing Intelligent Agent Systems: A Practical Guide*. West Sussex, England: John Wiley & Sons Ltd.

Passin, T. B. (2004). *Explorer's Guide to the Semantic Web*. Greenwich, CT: Manning Publications.

Wei, R., X. Zhang, L. Ding, H. Ma, and Q. Li (2009, October). A knowledge based agent prototype for Chinese address geocoding. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, Volume 7146, Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference.

Wooldridge, M. (2002). *An Introduction to MultiAgent Systems*. Chichester, U.K: John Wiley and Sons.

Zandbergen, P. A. (2008). A comparison of address point, parcel and street geocoding techniques. *Computers, Environment and Urban Systems* 32(3), 214-232.