MY TIS: THE ARCHITECTURE

Gomes Rocha, Nuno M.[1,2]

[1]*Faculty of Engineering, University of Porto, Portugal;*

[2] *PROEC – Projectos, Estudos e Construções Lda., Braga, Portugal*

ngomesrocha@gmail.com

ABSTRACT

This paper presents myTIS software architecture (TIS excerpt stands for Traveller Information System). It is a computational application to help commuters and non-commuters to plan their daily trips in public transport (PT) modes. In an intuitive environment, assists PT passengers during their displacements. This application is running in smart/mobile phones and in a web server.
It is clear that sustainable city development is going to be based in effective PT networks. Already in some developed countries, different transport modes are encouraged not only because of sustainable development issues but mainly to protect the environment. This is pressuring mass transport companies to increase quality of service (QoS). PT is expected to reach door-to-door level, to be a viable private car substitute. Such milestone requires improved communication between companies, rolling stock and passengers.
Our conclusion, after comparing private and mass transport modes, is that the key point for a substantial modal shift is in public transport inter-modality optimization at urban, regional, national and global level.
Transport networks complexity, for example bus networks, is in most cases a good excuse to prefer private modes. So myTIS main focus is to become a powerful tool in decision process of a mobility act for an individual centring it in public transport.
It is expected to be also useful for lower mobility or handicapped groups by providing them important assistance during their daily trips.
myTIS architecture is a multi-agent based solution forming a triangle. In base vertexes are mass transport companies and rolling stock, managing and inputting data to the system. In top is passenger, trading information with the system.
This paper will start with a featuring presentation that will be followed by potential architectures analysis. Super-structure presentation will precede a sub-divisions overview. Future work and conclusions will finish it.

1. INTRODUCTION

Cities are epicentres of societies' social interaction. As explained by Bettencourt et al. (2008) since civilization beginning, cities are related with humanity development. Such development brought human specialization into peoples' life. However specialised person and job geographical coincidence is limited. This fact generates mobility that impacts land-use options and urban areas development.

But mobility generation is also associated with personal desires satisfaction. From basic to sophisticated everyday life actions, almost every Human activity is creating displacements. Even with Internet services increasing, every activity is generating peoples/goods movements. Between other reasons, planning mistakes of such mobility needs causes overcrowding.
Similar circumstances have similar impacts in other dynamic system. Electrical systems, for example, shut down in overcrowding conditions. However in a dynamic system like a city shutting down is not an option. Because of different agents' heterogenic characteristics, such option will generate a diverse choices/options pattern and may be chaos.
Complex networks sustainability and management is only possible if selected solution considers problem different variables.
Modern cities mobility paradigm is based on "personal needs, private ways" metaphor. This is leading to mobility infrastructures saturation with impacts in urban areas life-quality.
Authorities' common approach, for this problematic, is increasing infrastructures capacity. However these actions just relocate or delay effective solution. A sustainable solution has to consider volume reduction/optimization and improved land-use.
In order to achieve cities and societies sustainable social and economical growing, mobility paradigm will be transformed into "personal needs, common ways".
To understand this it is important to characterise private mobility increased value. It is based in three aspects: availability, comfort and capacity to be a social status regulator. In our perception, this means that common mobility will have to deal with it in two different perspectives: information availability/communication and QoS.
A potential solution comes from interoperability and inter-modality optimization. Despite investments, public transport has not reached door-to-door quality standard as yet.
So the solution is in an improved intercommunication between passengers and PT services. This communication has to be focused in two critical points: approximate both sides' expectations and abilities.
Today information systems encompass a huge range of formats that ease communication and knowledge sharing. It ranges from advanced traveller information systems (ATIS) or classical stop timetables to online journey planners (OJP).
In this paper is presented a multi-agent based architecture. myTIS is considered a mixed approach between an ATIS and an OJP.
According to Gomes Rocha et al. (2009) it is a project in development providing passengers with a multi-modal public transport route advisory, with web interfaces for PC and mobile interfaces for PDA and similar portable devices.
Main features include O-D matrix development, "wishes" mobility factors for destination inputs, agenda planning, a ticketing system and adapted route. Crawford (2008) supports and explains importance and potential of tools that include some of these technical characteristics.
Such complex structure presents various implementation levels. From database structure to coding, every aspect is considered during software architecture development phase.
Because myTIS is designed to be a multi-modal public transport network planner, an object database model is used to conceptualise transport networks at local, urban, regional and global level.
Programming language is C#, for deeper levels, and various others, including Java™ for superficial layers.
Such scenario is inherently complex and due to its many autonomous decision-makers, it bears a great uncertainty nature. Multi-agent systems are specially well-suited to tackle such problems. Entities heterogeneity, social abilities and both personal and social goals pursuing representing complex domains are objective problems for such architectural solutions.
Without mortgaging future developments, associated with technical development, used framework for development is Windows Mobile™ Compact dot-Net Framework 3.5 mainly because of its interoperability, allowing access to other functionalities implemented in

different environments, and Common Language Runtime, reducing constraints about specific CPU capabilities in which application could run.
Database management system is PostgreSQL because of powerful add-ons not only for geographical objects, but also for shortest path algorithms implemented in Post GIS.
Cloud computing is being considered as a good solution especially because data complexity for smart/mobile phones processing capacity.
A critical aspect is to choose most operable shortest path algorithm. Dijkstra algorithm, with its variations when implemented with approximate buckets or with double buckets, is considered faster, in comparison with graph growth algorithm implemented with two queues. While last one suite perfectly with one-to-all shortest path problem, Dijkstra can be terminated if goal is a one-to-one or one-to-some path. Application will run differentially according to problem goal.
Architecture superficial layer forms a triangle between three agents. Passengers, rolling stock and public transport companies are three entities responsible for system operability.
This solution has potential to have considerable social impact, especially by contributing to modal exchange, affecting traffic condition in major urban areas. It is expected to draw commuters and non-commuters attention back to public transport as many of its disadvantages can be minimised, if not surpassed, as users will have access to accurate and up-to-date information about trip options whenever a mobility decision is to be made. An important impact is also expected in handicapped or lower mobility groups.
This paper will be followed by a features review before similar applications and potential architectures are presented. Sub-modules will precede super-structure presentation. The paper will be concluded after future work is discussed.

2. BACKGROUND

2.1 Web server

In web server, passengers are expected to use pre-trip planning. “Origin and destination input” and “trip planning”, “historic” search and “timetable” consultation is available.
A personal “profile” with mobility information could be created. Data quiz for mobility aspects, age and other factors affecting walking speed, is available. This data will be transmitted to mobile device for trip planning.
After profile creation, “agenda planning” is operable. Passenger could save daily tasks, expected duration and location. With this data a daily trip plan in public transport is generated. All this data is saveable and recycled for future use. “Agenda trip plan” is updated in mobile device. This feature is expected to have major impact.
It is also possible to buy multi-modal public transport tickets. Web server is expected to be used has a base camp; however it is totally synchronized with mobile device.

2.2 Mobile devices

Passengers are expected to use mobile device application to en-route re-planning. Passengers have access to basic “trip planning” for direct use or for friend sharing in small text message or other formats files.
Origin input is possible by texting, map selection or automatically generated by geo-reference like GPS, Galileo or similar solutions as presented by Hamet (2009).
Destination input is made in similar ways, texting and map selection. However for geo-reference origin input, destination could be inputted as a “personal wish”. In this situation passenger could, with voice control or by texting, input what desires. This option is expected to be useful for tourists arriving unknown cities.

Planned trip output will be sonorous, graphic or both. Timetable consultation and historic searching is also available.
En-route re-planning is four factors result: public transport availability, personal profile characteristics, geo-reference location and rolling stock information. An optimized trip plan is generated inputting rolling stock arriving time, passengers' available places, passengers walking travel times and preventing delays and overcrowding situations.
Information different origins, such as web server data centres or mobile device, make cloud computing as a suitable solution.
"Geo-reference tracking" is available not only for basic trip plan but especially for "agenda planning". This feature is important in case some premise in trip plan changes. If location or task duration is about to be exceeded, or even if passenger misses a modal change, system will automatically send alerts and recalculates trip plan and its entire links, namely tickets and payments.
Another important point is "route recognition". If application do not recognises a route/network line in its database, passenger is asked basic information about it. This feature recognises an important trend in computational science and societies in general: collaborative networks. It is expected to be an important feature during initial stages of market implementation process.
In mobile device is possible to buy multi-modal tickets and use it as a ticketing system.

2.3 Back office

Back office has two different entities interacting in it: "system management" and "public transport companies".
"System management" team is responsible for interaction between external entities and myTIS. For example with internet banking for ticketing service; or public transport companies for data sharing.
For example, using historic search, for specific region or network, an O-D matrix is generated. Data mining and data fusion tools will use such information to infer and predict transport opportunities, overcrowding, misusing or other unexpected effects in PT networks. This information is useful for approximate passenger expectations and PT capabilities.
"Public transport companies" also have access to back office. Tools like add, change or remove a PT lines are at their disposable. Public transport companies have the best interest to maintain these areas updated because passengers will have access to an evaluation system that gives chance to evaluate and quantify QoS of each company creating positive market concurrence.
This evaluation process is a growing trend in users and Internet services interaction. People like to be heard and if there is an immediate way to do it, better.

3. SOLUTIONS

In architectural development, several stages must be passed. Similar solutions must be study, database structuring must prevent future intolerabilities, different options must be considered, technical and organizational aspects must be guaranteed. Following it is presented myTIS architectural development process. Some aspects are generally presented while most important issues are thoroughly explored.

3.1 Similar applications

Similar applications were studied in two dimensions. On one hand are OJPs' and other applications in web portals and on the other hand are ATIS and other generic transport applications.

3.1.1. Web Portals

Some of most popular OJP web portals studied present questions divided in three areas: input requirements, inter-modality and output quality.
Most services are highly exigent in passenger knowledge requiring different types of inputs.
This situation is comparable to Internet search engines evolution. First generation knowledge requirements were much higher than second or third generations. This shows that input requirements reduction is a tendency in computational systems evolution process. Such fact was considered in myTIS architectural development.
In most cases inter-modality is low, especially at regional and national level. Most public transport companies explore niches without considering “all-together” potential. This happens because private modes increased value is overvalued. Social and economical modern tendencies must be analysed to explore private modes vulnerabilities. In our vision, this requires an improved balance between public transport companies internal and external concurrence.
In some cases output quality ratio with input requirements has to be considered low. This is a major issue in myTIS architectural development.
These three questions are closely associated with databases structure and management.

3.1.2. Transport networks applications

Presented applications descriptions are based on its patents founded in WIPO (World Intellectual Property Organization) databases.

3.1.2.1. Signing system

Mucke (1988) proposes an information system where stopping points are indicated by a numbered sequence which is selectively illuminated to indicate vehicle position.
A counter simultaneously provides a count value identifying next stop along route with a pointer indicating travel direction. Alternatively counter provides a count value representing last stop along route.
This solution is mainly used by buses guaranteeing that traveller information is separated from local information on signs along route.

3.1.2.2. Boarding point guide notification system

Nippon Denki (2003) suggests that passenger inputs PDA travel and positional information with respect to base station to a service server. Then server extracts guide information corresponding to receive information from a database and transmits it to PDA through a switching system.
Such solution is useful as a boarding point guide notification system to provide guide information about destination, in real time, corresponding to travel information. This enables traveller to reach destination comfortably.

3.1.2.3. Personal Navigation System

Di Capua (2003) presents a system that includes a data processing centre, composed of a data processor and receiving/transmitting signals device, a multiplicity of personal use devices or free-ways, a control and energy autonomy devices and a various position sensors.
Storing device, in processing centre, stores a PT network representation and timetables that serves it. Meanwhile data processor is programmed to process path information.
This system allows passengers inside public transport networks to found a path from a place to another, in portable ways and can be used always and anywhere, as explained by Di Capua (2003).

3.1.2.4. Travel information system

Closer approaches from myTIS are proposed by Fischer (2003) and Ficher et al. (2005). However architectural designs are different. As it is possible to infer, both solutions are based in central controlling infrastructure while myTIS is dispersed.
Fischer (2003) presents a solution where a destination is inputted to mobile device and transmitted to a master computer in which a location is assigned to mobile device. A master computer has connections to public transport control devices and receives vehicles current stopping locations.
Based on these locations, an individual timetable of suitable PT vehicles to reach destination is defined. Timetable is transmitted to mobile communication device and reproduced visually or audibly.
System biggest advantage is that passenger do not need to know is current location to efficiently receive travel plans to reach its final destination.
In a similar current Ficher et al. (2005) proposes a passenger travel information system that is used in conjunction with a central infrastructure in order to provide individual personalized travel information.
Personal device has inputs for querying and obtaining travel information and communications for duplex communication with a central infrastructure. This includes services provided by an intercity and local public transport services.
This innovation ensures that passenger is provided with up to date comprehensive information during his actual travelling. Passenger is provided with a virtual travelling companion.

3.2 Architectures

Previously presented solutions are based on central structures. In our perspective, such infrastructures present constrains in querying process. Even in decentralised approaches these problems arise.
Some database skeleton characteristics are essential to avoid this problematic. Those characteristics are presented bellow.
Multi-agent architecture applicability is suggested for complex scenarios. Transport networks with different integrators with social abilities (drivers, pedestrians and travellers) seams the perfect scenario to test it. However other designs must be considered. The analytic process of studied designs is presented in this chapter.

3.2.1. Database structure

myTIS database structure is divided in two areas: data structure and geographical coordinate system.
Data structure includes public transport network conceptualisation and data correlation. Main issue is to guarantee querying linearity.
Geographical coordinate system is used by two important agents: passengers and rolling stock. Critical aspect is different agents' synchronization.

3.2.1.1. Data Structure

Data structure is composed by layers. Layers are divided in units. Units are then sub-divided in levels. Levels are composed of tables that could be divided according five attributes: "geometry", "additional attribute", "extended attributed", "relational" and "index".
For example, to "Street Layer" are associated units like "Main Roads" and "Secondary Roads". These units are then divided in different levels like "Road Geometry" or "Traffic Restrictions". These levels are connected to respective tables.
An important layer is "Public Transport Stops Network". This layer is associated to others by relational tables. It means that a specific stop is associated with specific "Roads", "Transport Mode" or "Mobility Factor".
"Catchment Areas" are defined for each "Public Transport Stop". "Areas Dimension" is variable to transport mode and it is conceptualised in a geometry table.
For example, "Airports" serves regions while a "Bus Stop" is associated with some streets. This means that even if a region/street could be geometrically (inside mobility circumference) associated with a "Public Transport Stop", this option is only validated if there are no "Physical Limitations" (no crosswalk in street or other).
This aspect is essential to implement linear querying and represents a differentiated approach from previous solutions.
Another example of database structure importance on output quality is presented in "Wish Planning" feature. Such feature requires relational tables between layers like "Points of Interest", "Roads" and "Mobility Factor".
Data relation is critical to information quality. Relational tables are essential to approach public transport door-to-door level.

3.2.1.2. Geographical coordinate system

Geographical coordinate system is based on five parameters: "Map Projection", "Cylindrical Projection", "Supported Geodetic Data", "Supported Spheroids" and "Units".
Map projection parameters include elements like "Scale factor", "Azimuths", "Centre Longitude" or "Latitude" and "Origin Latitude" or "Longitude".
"Centre Longitude" and "Latitude" will define "Map Projection" centre point while "Origin Longitude" and "Latitude" will define origin x and y-coordinates respectively. These elements establish a relation between "Map Projection" and location and avoid distortions. In table 1 is possible to find some parameters included in map projection.

Table 1 – *Some map projection parameters*

Parameter			
azimuth	False-easting	Path-number	Longitude-of-origin
Scale-factor	Central-meridian	Landsat-number	...

"Cylindrical Projections" parameter establishes, for "Supported Map Projections", relational elements between them and "Pseudo Cylindrical Projections". In table 2 is possible to find some of these relations.

Table 2 *– Some cylindrical related to pseudo cylindrical projections*

Cylindrical	Pseudo cylindrical
Cylindrical equal area	Eckert II
Behrmann	Craster parabolic
Miller cylindrical	McBryde-Thomas flat polar quartic

"Supported Geodetic Data" establish object relations between geodetic information with different origins.
"Supported Spheroids" associates elements with semi-major axis and inverse flattening.
Last parameter is "Units". It could be divided in "Linear" and "Angular" units. Both units are associated with conversion factors according to specific area on earth. These parameters are essential to define shape and earth size as well as reference point for various coordinate systems used in earth mapping. In table 3 and 4 is possible to find some units used.

Table 3 *– Some linear units' conversion factors*

Unit	Conversion factor
Meter	1.0
U.S. Foot	12/39.37
Indian Foot	12/39.370141
Fathom	18.288

Table 4 *– Some angular units' conversion factors*

Unit	Conversion factor
Radian	1.0
Decimal Second	(∏/180)*3600
Grad	∏/200

3.2.2. Software architecture

Bass et al. (1997) establish that software architecture is informatics system structural elements and its relationships.
System could be defined in two different visions: UML 1.3 and IEEE Std.610. Despite its similarities, both visions have impacts in architectural design. These two different visions are resumed bellow.
Another important aspect is to define system critical characteristics. Those characteristics are presented in this chapter.

3.2.2.1. UML 1.3

In a UML perspective, a system is a collection of connected units that are organized to accomplish a specific purpose. A system can be described by one or more models, possibly from different viewpoints.
Architecture can be recursively decomposed into parts that interact through interfaces, relationships that connect parts, and constraints for assembling parts. Parts that interact through interfaces include classes, components and subsystems.

It is presented bellow a common definition used by UML evangelisers.

> *Software architecture of a program or computing system is the structure or structures of the system. It comprises software components, externally visible properties of those components and relationships among them. By "externally visible" properties, we are referring to those assumptions other components can make of a component, such as its provided services, performance characteristics, fault handling, shared resource usage, and so on. The intent of this definition is that a software architecture must abstract away some information from the system (otherwise there is no point looking at architecture, we are simply viewing the entire system) and yet provide enough information to be a basis for analysis, decision making, and hence risk reduction.*

3.2.2.2. IEEE Std. 610

IEEE defines that a system is a components collection organized to accomplish a specific function or set of functions.
An IEEE definition is coated bellow.

> *Software architecture is components structure of a program/system, their interrelationships, and principles and guidelines governing their design and evolution over time*.

3.2.2.3. Architecture decisions

Malan et al. (2002) says that key architectural decisions should be focus on systemic and high impact (important to business) issues. This includes: “Priority Setting”, “Decomposition and Composition”, “Properties” (especially cross-cutting concerns), “Fit to Context” and “Integrity”.
These aspects were considered in myTIS architectural design.

3.2.2.3.1. System priority setting

It is essential to make priorities explicit so that attention can be focused on high-priority areas. Trade-offs between conflicting concerns can be made rationally, and decisions can be justified in agreed priorities terms.
Priority-setting process for system technical aspects must be considered as a highly strategic process considering five points:

1. Business, including business strategy and direction, core competencies and resources, and politics;
2. Market including customers, competitors, suppliers and channel;
3. Technology including trends and opportunities;
4. Constraints including existing technology investments and legacy systems;

5. Challenges and impediment to system success, development and business.

These aspects affected myTIS feature division. To establish featuring a content matrix was developed with special focus on “priority” and “complexity”. A relational pattern was established considering previously presented aspects. In table 5 is presented a matrix overview.

Table 5 *– System priority matrix*

ITEM	Modal division	Relations	Involved Agents	Priority	Complexity	Description
X	X	X	X	X	X	X
X	X	X	X	X	X	X
X	X	X	X	X	X	X
X	X	X	X	X	X	X

3.2.2.3.2. Decomposition and composition

Define primary system structural elements or components, and their interrelationships is fundamental.
Concerns as “Complexity” (applying “concerns separation” and “divide and conquer” principles), “Portability” and “Flexibility” (applying principle of localizing areas that are likely to change together) arise during architectural development.
It is expected that system functionality or services can be delivered by components working in collaboration. Such problematic has to be done from a system perspective, so that consistent assumptions are made about each component responsibilities.
System decomposition isolates some particular concerns so that they can be addressed independently. Different partitioning choices tend to isolate different concerns sets.
These aspects are connected with “Modal division” and “Relations” in matrix presented in table 5.
“Modal division” includes areas like: “Inputs”, “Planning”, “Historic”, “Profile”, “Ticket buying”, “Configuration Panel” or “Map visualization”.
“Relations” camp defines intercommunication process between different areas of modal division. It also includes implementation process for each element.

3.2.2.3.3. System properties and Cross-Cutting concerns

Concerns impacting various system parts should be considered. At a minimum, sets of collaborating components have to be considered together, to properly address each cross-cutting concern.
“Performance”, for example, typically has to be considered in interaction patterns that architectural structure allows. It is not just a matter of separate parts performance optimization.
Internal mechanism must be designed as a collaborating components set, and in particular, those components specific roles. It has to be focused on addressing cross-cutting concerns.
For example, result may be to place an interface on a component that does not fit with its core cohesive responsibilities.
These aspects are included in “Relations” and “Description” camps in table 5 matrix.

3.2.2.3.4. System fit to context

Key technical considerations alluded to this aspect has to do with “Interoperability”, “Consistency” and “Interface” with external systems.
Fit within development organizations’ culture and capabilities, are also considerations to be factored into architectural decisions and choices.
These aspects, presented in table 5, are considered in “Involved Agents” abilities and “Priority” levels.
Priority relational pattern is established by 1-5 punctuation. Higher priorities were given to existing features in other applications. This process was established to reach existing market level. Such strategic positioning proved useful to define a work base.

3.2.2.3.5. System integrity

“System Integrity” means having, or being conceived as having, a unified overall design, form or structure. It has to do with “Pieces Congruence” in large, as well as in small scale.
Such aspect integrates all camps shown in table 5 and it is a validation process before implementation.
While matrix development is made from left to right starting with “Item” definition, validation process is made from right to left. Objective is to create a comprehensive “Project Tail”. If this aspect fails it is better to restart process.

3.2.3. Software architecture models

After studying similar solutions, define a development vision, establish a database structure and define system characteristics, it is important to study potential architectural solutions. Selection process is based in two steps: “Exclude” and “Focus”.
According to presented premises five models were studied: “Blackboard”, “Client/Server”, “Multi-agent architecture”, “Distributed systems” and “Service-oriented architecture”.
An overview of four exclude designs characteristics is shown in next points.

3.2.3.1. Blackboard systems

According to Corkill et al. (1986) blackboard systems basic components are: “Blackboards”, “Knowledge sources” and “Control component”.
“Blackboard” is a global database containing input data, partial solutions, and other data that are in various problem-solving states.
“Knowledge sources” are independent modules that contain knowledge needed to solve problems that can be widely diverse in representation and inference techniques.
A “Control component” makes runtime decisions about problem course solving and problem-solving resources expenditure. “Control component” is separate from individual “Knowledge sources”. Some “Control Component” characteristics do not fit with myTIS requirements.
In some blackboard systems, “Control component” itself is implemented using a blackboard approach. Blackboard metaphor is presented as:

> *Imagine a group of human specialists seated next to a large blackboard. Specialists are working cooperatively to solve a problem, using a blackboard as workplace for developing solutions. Problem solving begins when a problem and initial data are written onto blackboard. Specialists watch it, looking for an opportunity to apply their expertise to developing*

solution. When a specialist finds sufficient information to make a contribution, it records it on blackboard, hopefully enabling other specialists to apply their expertise. This process of adding contributions to blackboard continues until problem has been solved.

3.2.3.2. Client/server architecture

Lehmann (1970) says that this architecture model is based on “Server”, that offers a service, and a “Client”, which requests a service.
A “Client” can be characterized by sending a message to a “Server”, requesting it to perform a particular task, the service.
Clients usually represent user interface portion of an application, allowing users to get in contact with and use underlying application. Clients are responsible for validating data entered by user, transmitting requests to “Server” and, in some cases, even executing business logic on received results. Clients’ number accessing a server is not limited per model definition.
“Server” is a component in client/server architecture, which receives clients’ requests. It aim is to solve requested task and send result back to client. Solving these tasks can mean implementing complex program logic, accessing database systems and managing large data resources. Client/server interrelation does not satisfy our communication needs.
Result should contain an answer to clients’ request. Solving a clients request does not have to be done by server, which originally received request. Server itself rather could be connected to other servers.

3.2.3.3. Distributed systems

An important goal of a distributed system is to support sharing, exchange, and resources/information management such that implementation aspects are transparent to its users as Homburg (2001) supports. Message passing, shared memory abstraction or on remote objects are some paradigms used to develop distributed systems.
Wide-area distributed systems have to deal with a variety in services, functionalities, and interfaces for underlying operating systems and networks. At the same time they should hide these differences from their clients, users or applications, leading to different implementations for similar components.

3.2.3.4. Service-oriented architecture

Ort (2005) suggests that “find, bind, and execute” paradigm allows service consumer to ask a third-party registry for service that matches its criteria. If registry has such a service, it gives consumer a contract and an endpoint address for service.
This architecture is based on six entities: “Consumer”, “Provider”, “Registry”, “Contract”, “Proxy” and “Lease”.
Service consumer is an application, service, or some other type of software module that requires a service. It is an entity that initiates service locating in a registry, binding to service over a transport, and executing service function. It executes service by sending it a request formatted according to contracted.
Service provider is a network-addressable entity that accepts and executes requests from consumers. It can be a mainframe system, a component, or some other type of software system that executes service request. It publishes its contract in registry for access by service consumers.

Service registry is a network-based directory that contains available services. It is an entity that accepts and stores contracts from service providers and sends those contracts to interested service consumers.
Contract is a specification in service consumer interaction with service provider. It specifies request format and response. A service contract may require a set of preconditions and post-conditions. Preconditions and post-conditions specify state that a service must be in to execute a particular function. Contract may also specify QoS levels.
Service provider supplies a service proxy to service consumer. Service proxy can enhance performance by caching remote references and data. When a proxy caches a remote reference, subsequent service calls will not require additional registry calls. By storing service contracts locally, consumer reduces network hops number required to execute a service.
Service lease, which service consumer registry grants, specifies valid contract time amount. Without lease notion, a consumer could bind to a service forever and never rebind to its contract again. This would have effect of a much tighter coupling between service consumer and service provider.
This architectural design will increase our data requirements without improving output quality.

4. SUPER-STRUCTURE

An agent is an encapsulated computer system situated in an environment and capable of flexible, autonomous action in that environment in order to meet its design objectives as Jennings (2001) explains. In multi-agent systems, agents have characteristics of autonomy, limited perception and diffusion. This means that each agent is at least partially autonomous, does not have capacity to use all information and there is no controller agent. These are major differences between myTIS architecture and similar applications.
Agents could be reflexive or learners. Russell et al. (1995) says that reflexive agents are those that work only if correct decision can be made on current percept basis. Learners will run critical environment evaluation response in order to improve its performance element. In this system some agent are going to be reflexive and others learners.
In multi-agent systems, agent concept includes not only computational systems but also human agents.
Another important aspect is “Game Theory” application to multi-agent systems. This aspect studies interaction between self-interested agents as explained by Parsons et al. (2000).
“Game Theory” classical question is: What is best and most rational thing an agent can do? Each agent must take into account decisions that other agents may make, and must assume that they will act so as to optimize their own outcome.
However most compelling “Game Theory” applications to multi-agent systems have been in negotiation area that in this case means the process by which agents can reach agreement on common interest matters.
These are some agent characteristics that could be found in super-structure.
System main agents are “Passengers”, “Public Transport Companies” and “Rolling Stock”. Other agents include “Maps”, “Public Transport Database”, “Public Transport Directory”, “Ticketing”, “Data Recycler” and “Transport Engineers”. Smaller agents include “Public Transport Stops” or “Banking”.
“Passenger” interacts with three other agents: “Mobile device”, “Web Server” and “Public Transport Stops”.
“Mobile” and “Web Server” are used to trip planning and ticket buying process. They are synchronised to share data between each other. They are both connected to “Data Directory” to trade information with it.

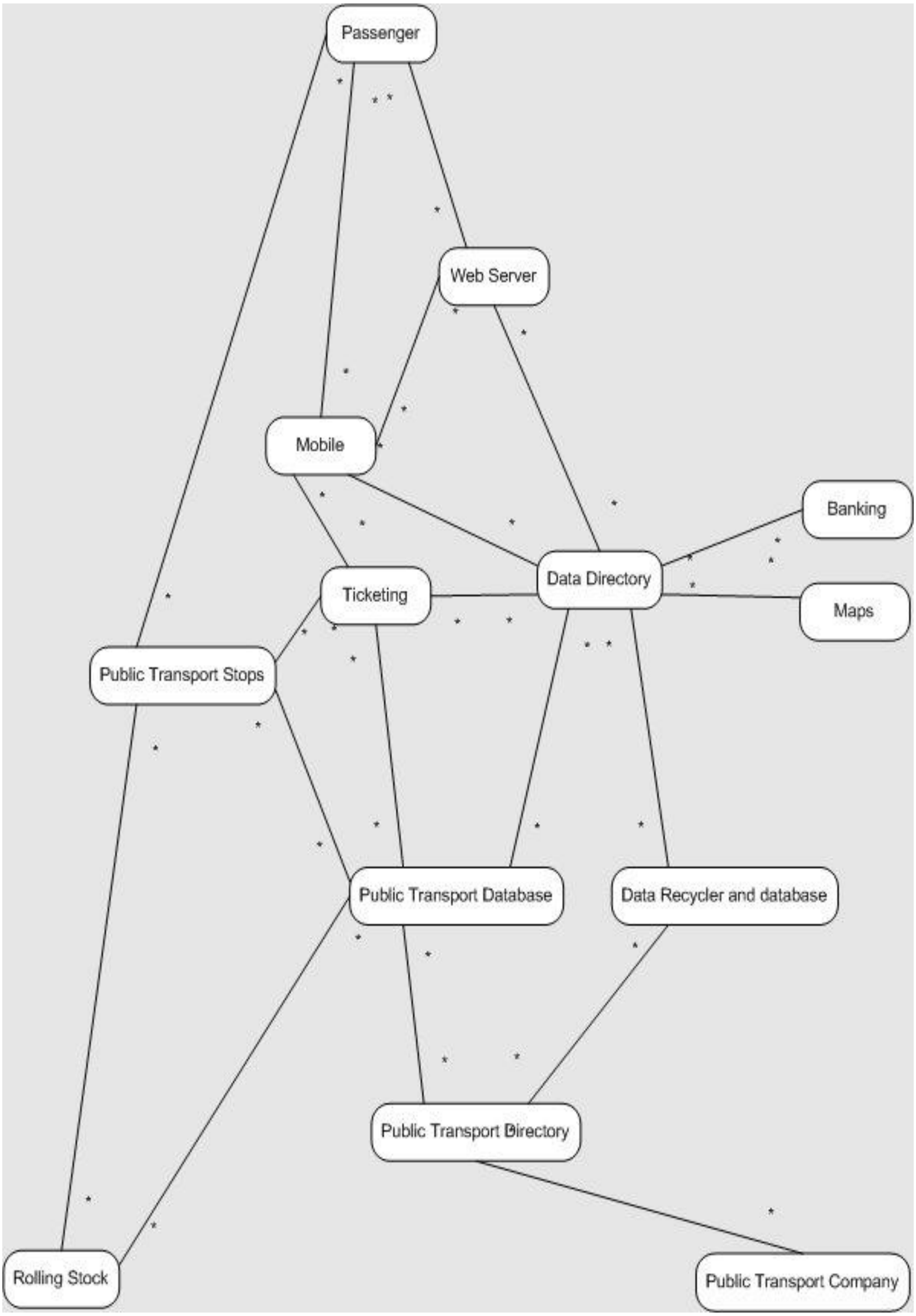

***Figure 1** – Super-Structure agents and interconnections*

"Passenger" interaction with "Public Transport Stop" is important to validate tickets and to generate data about "Public Transport Situation". This fact will affect other passengers' network interaction.
"Rolling stock" main interaction is with "Public Transport Stops". It is generating data included in "Public Transport Situation" that also affects passengers' options. Objective is to optimize transport capacity usage without creating overcrowding or other problems.
"Public Transport Companies" are generating data that supports all system and receives important information from "Passengers" and "Rolling Stock". myTIS supports companies in "Fleets management" and to improve and approximate their offer to passenger expectations.
Presented architecture shows a new solution to trip planning problem. This innovative approach could be divided in two areas: featuring and design.
Features like "Agenda" planning, "Multi-modal Transport planning", "O-D matrix" development or even "Wishes box" represent new approaches to the problem.
Some other features like "Updated information" or simple "Trip planning" are simplified and improved by new design.
Comparing to Granberg (2000) or Angermann et al. (1999) proposals it is possible to verify that design improvements suggest better and faster response to en-route trip planning.
Considering all previous facts it is possible to find on figure 1 an architectural proposal.

5. SUB-MODULES

Main agents' structure is presented bellow. For each agent (white balloons) is presented all available features (grey balloons) and intercommunications.

5.1 Passenger

In figure 2 is presented all features available for "Passenger". Main interrelations were previously described. As Human agent it presents unique characteristics that require specifications in system interaction process. As an essential agent, system operability, performance and interfaces are essential to success.

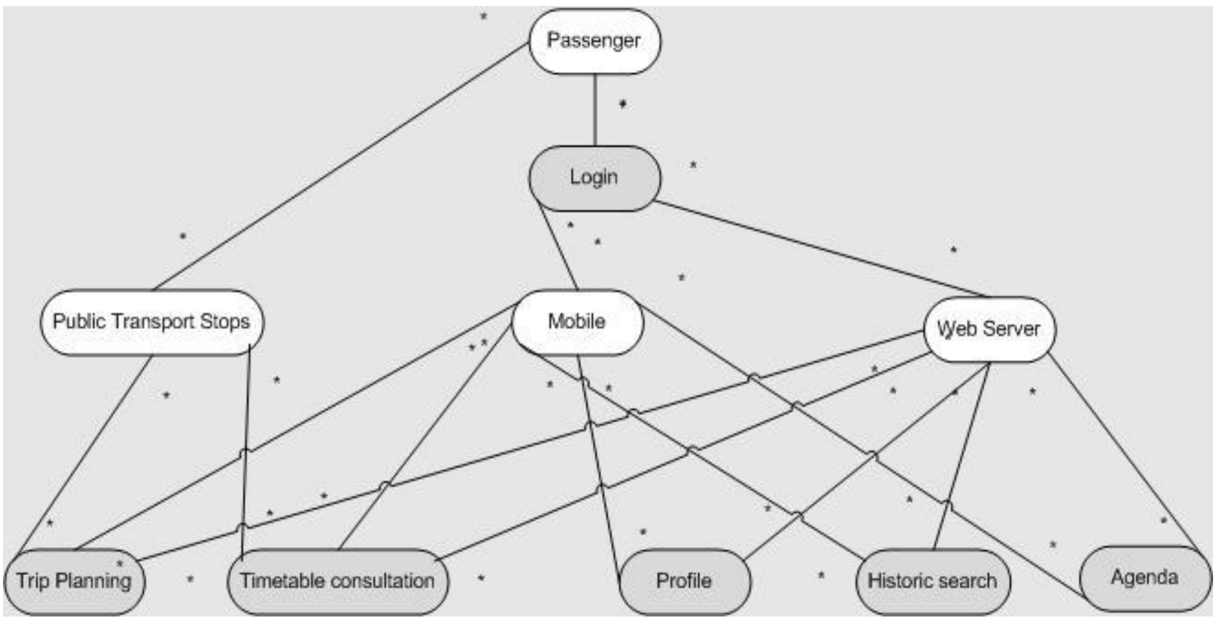

***Figure 2** – Passenger structure, as a system agent*

5.2 Public Transport Company

In figure 3 is presented all features available for "Public Transport Company". Main interrelations were previously described in super-structure. To success in the relation with this agent, it is important to balance shared data. This relation failure could be surpassed with collaborative passengers networks. It could be considered a Half-Human agent.

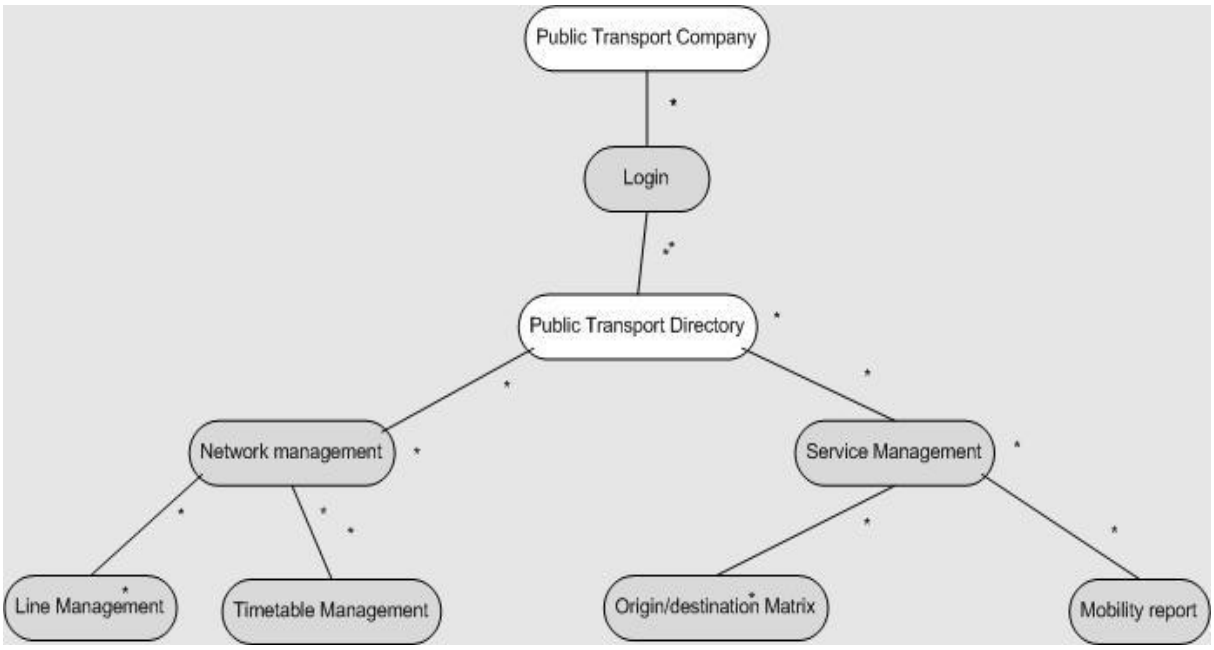

***Figure 3** – Public Transport Company.*

5.3 Rolling Stock

"Rolling stock" available features are presented in figure 4. Main interrelations were previously described. Main features are associated network management. In initial stages, this agent could be responsible for uncertainty, deregulation and inclusion of prediction measures. However this is expected to be surpassed in implementation process.

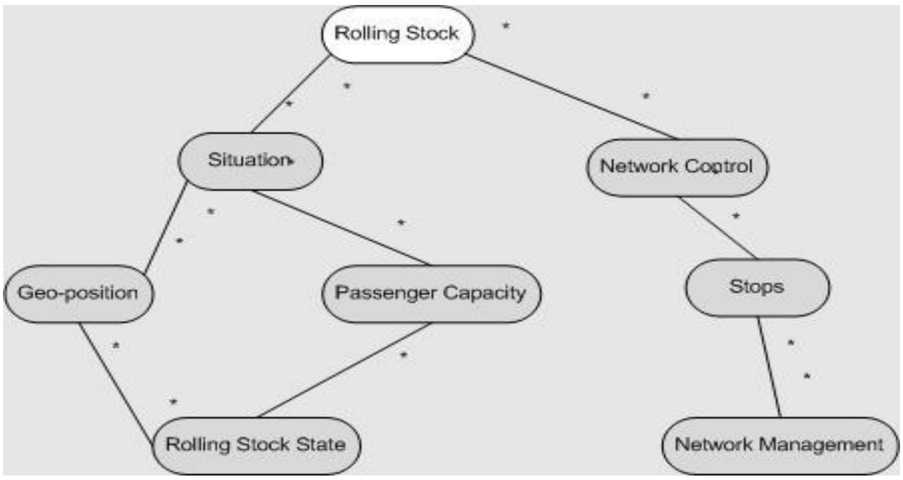

***Figure 4** – Public transport rolling stock structure, as a system agent*

5.4 Web Server

In figure 5 is available all "Web Server" features. Main interrelations with mobile device were previously presented. This agent could be used in three different ways, with or without login request. Initially to download mobile application, and later to PT information searching or to use personalised environment.

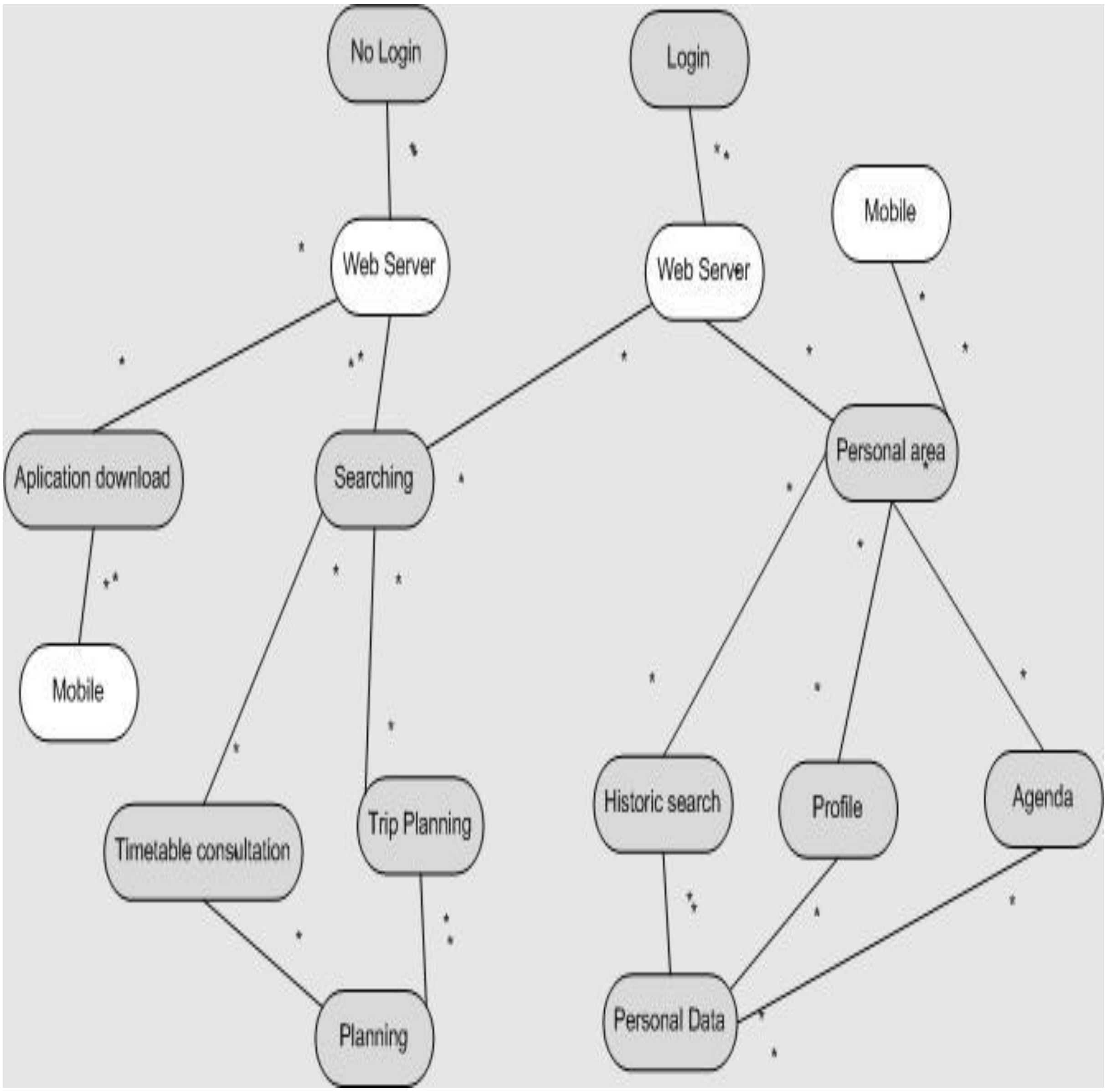

Figure 5*– Agent Web Server structure*

5.5 Mobile

In figure 6 is presented all features available for "Mobile" device. Main interrelations with "Web server" were previously described in super-structure. It is presented specific features like "Wishes box", "Agenda" planning (connected from "personal area"), "Route validation" (from "route detection") and "Ticketing" system. Some features require "Login" and others are open.

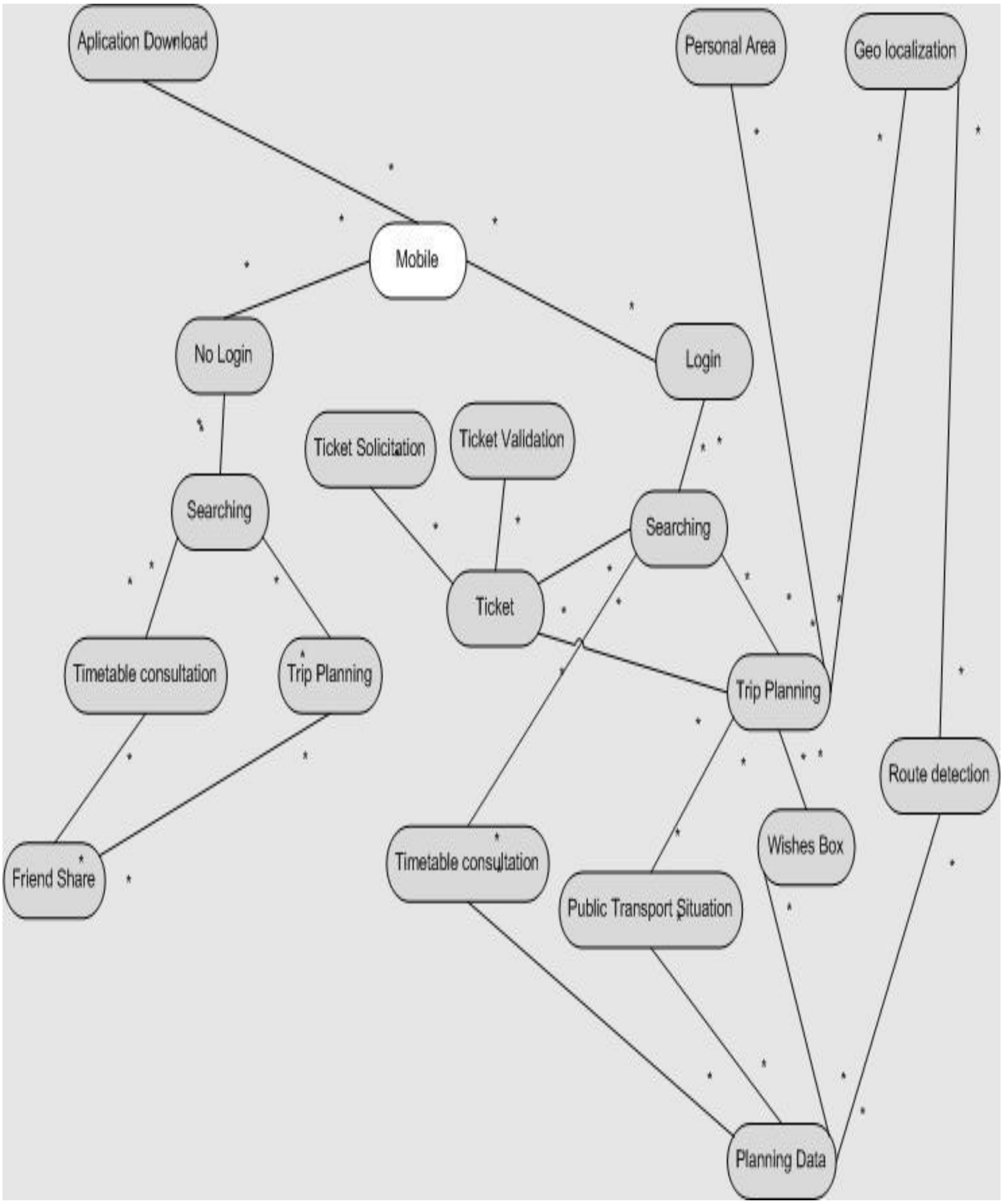

Figure 6– *Agent Mobile structure*

5.6 Data Directory

“Data Directory” features and interconnections are presented in figure 7. Main interrelations presented include: “External Institutions Manager”, “Storing” process, “Ticket management” process and “Rolling Stock Information” management process.

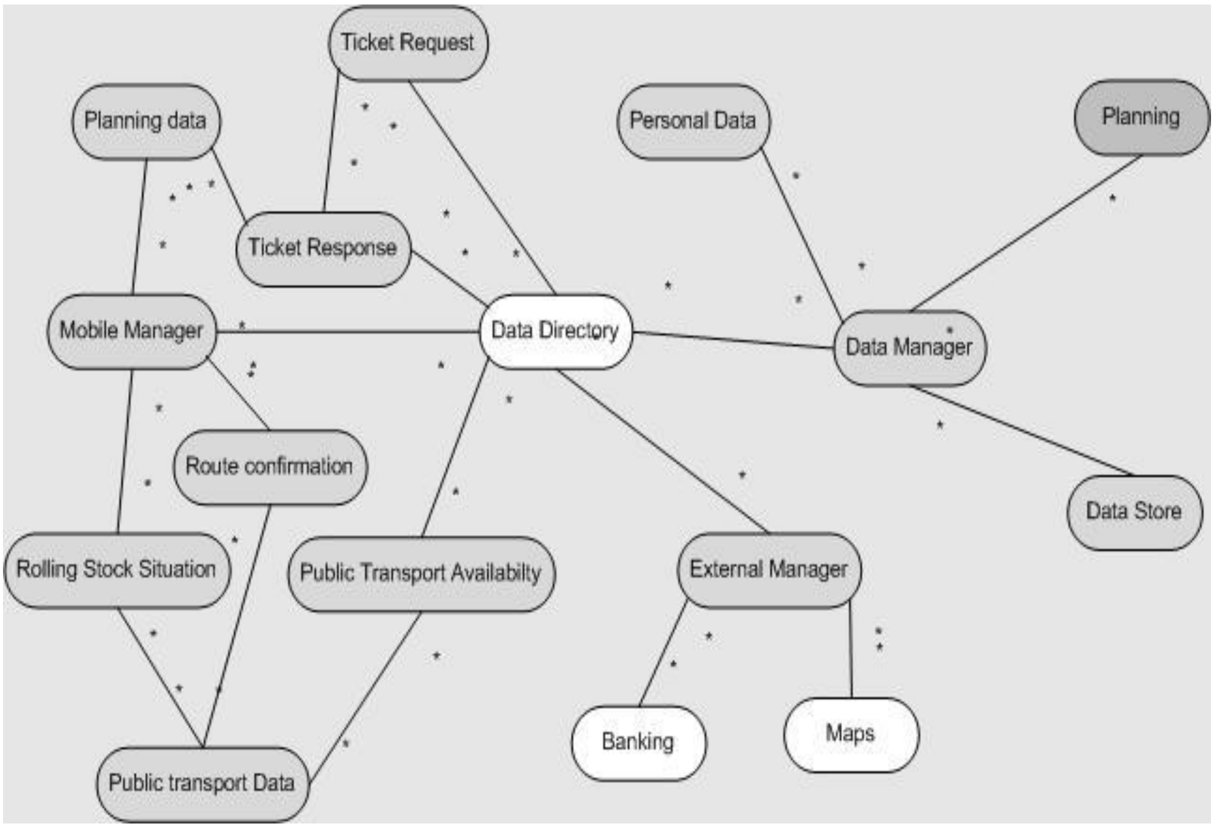

***Figure 7**– Agent Data Directory*

5.7 Ticketing and Public Transport Stops

In figure 8 is presented interrelation process between "Ticketing" system and "Public Transport Stops". This relation is critical to avoid overcrowding and improve QoS.

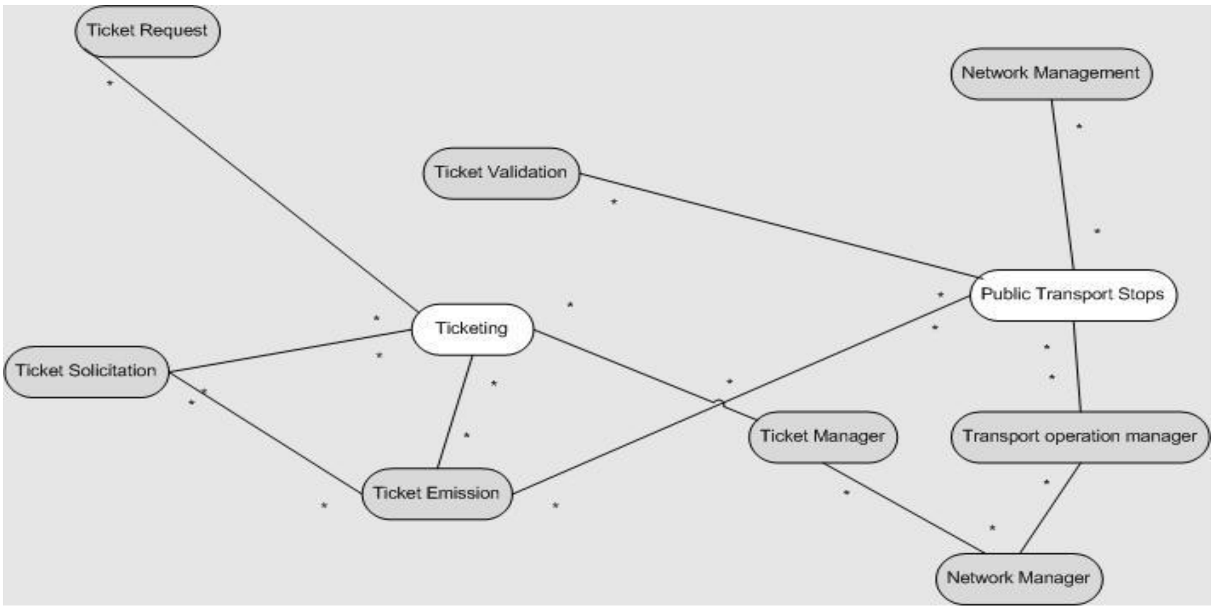

***Figure 8**– Ticketing and Public Transport Stops structures*

5.8 Public Transport Database

In figure 9 is presented all features available at "Public Transport Database". It establishes a support for public transport companies' fleet management and a base for trip planning.

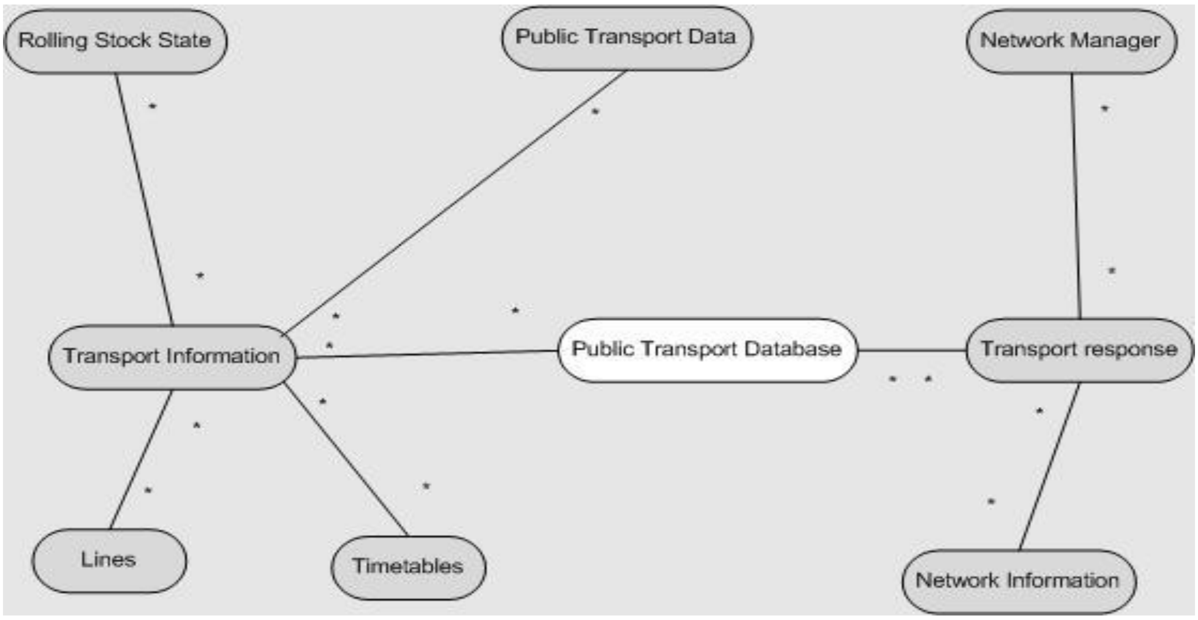

Figure 9– *Agent Public Transport Database*

5.9 Data Recycler and Database

In figure 10 is presented O-D matrix development in "Data Recycler and Database". It represents an important myTIS goal: approximate Human and Half-Human expectations and abilities.

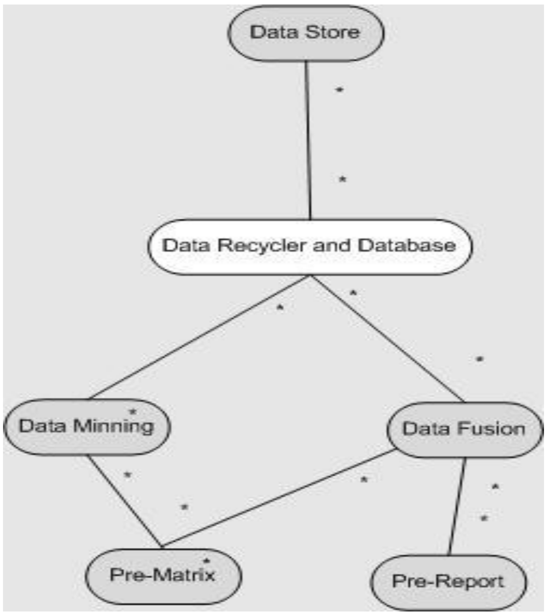

Figure 10– *Agent Data Recycler and Database*

5.10 Public Transport Directory

In figure 11 is presented all features available at "Public Transport Directory". It shows public transport companies data interaction and O-D matrix validation process.

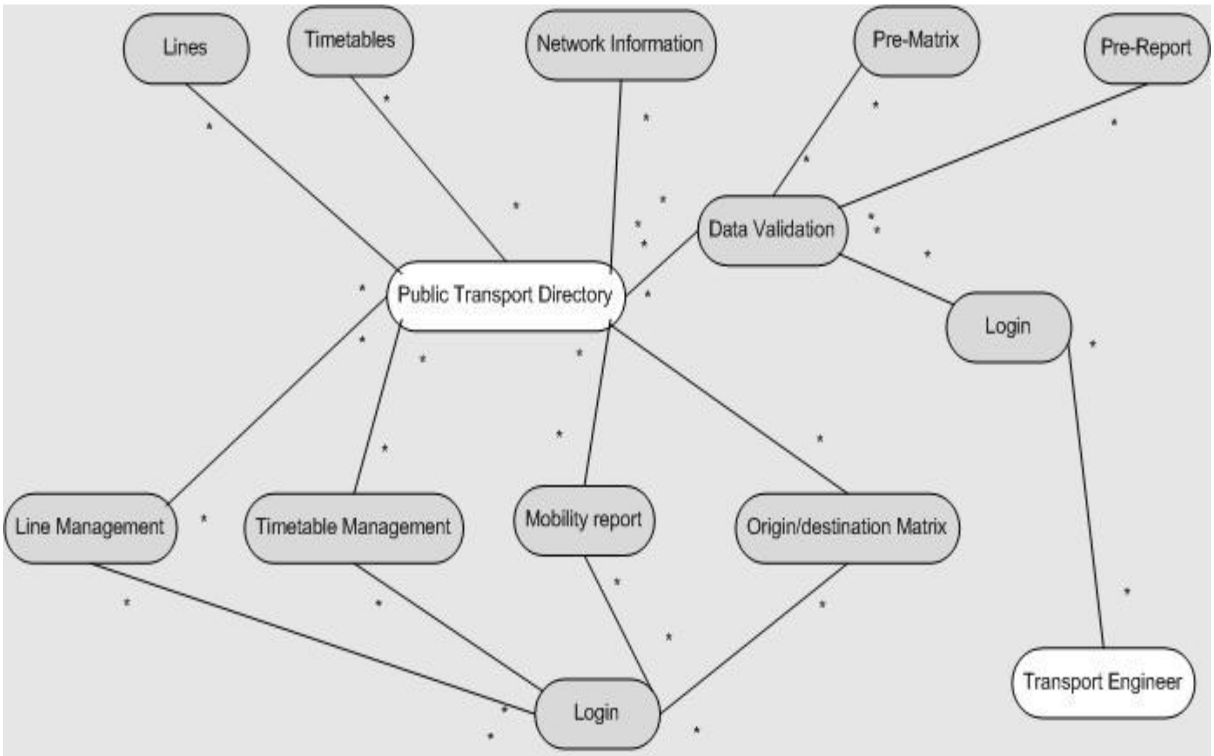

Figure 11– *Agent Public Transport Directory structure*

6. FUTURE WORK

Future work is divided in two phases. First phase includes a networking process for complete system implementation. After this phase is finished, initiate contact with companies in different market areas, such as mobile phone service, software producers, and public transport companies and so on, for market implementation. Some market implementation steps were previously described however complete model is still in study. These two phases are expected to be finished in five years.

7. CONCLUSIONS

Sustainable city growing is only possible based in competent public transport networks. In our vision future modal dispersion is going to be even between private and public modes. This will require public transport companies to achieve door-to-door level while public transport networks complexity expands. Such complex networks will require a higher level of communication between public transport companies and passengers. Scenarios like this are perfect environment for multi-agent systems. A multi-agent architecture is proposed to supply a system to help commuters and non-commuters, in a personalised environment, to plan and assist them during their daily trips in public transport modes. It is expected to run both in smart/mobile phones and in a web server.

Main system agents are passengers, public transport companies and rolling stock, maps, public transport database, public transport directory, ticketing, data recycler and transport engineers. System is expected to be on market in five years.

REFERENCES

Angermann, M., Steingass, A., Robertson, P., Hiller, W., Groth, T., (1999) Sistema para el suministro mejorado a un viajero de informaciones sobre vaijes, Patent Numbers: ES2211542, G06F 17/80

Bass, L., Clemens, P., Kazman, R., (1997) *Software Architecture in Practice (First Edition)*, Addison-Wesley.

Bettencourt, L.M.A; Lobo, J.; West, G.B., (2008) Why large cities are faster? Universal scaling and self-similarity in urban organization and dynamics, European Physical Journal B, 63, 285-293.

Corkill, D.D., Gallagher, K.Q., Murray, K.E., (1986) A generic blackboard development system, *Proceedings of the National Conference on Artificial Intelligence, 1008-1014.*

Crawford, D., (2008) Sure footed – looks at NYC Transit moves to make on-foot travel better informed, ITS International Magazine, Transmart, Nov./Dec,27-28.

Di Capua, F., (2003) Personal navigation system for public transport network, includes data processor able to process path information for the user, as well as able to compute and automatically debit travel and service costs, Patent Numbers: EP1326201-A;EP1326201-A1.

Fischer, M., (2003) System providing public transport travel information on mobile device, by retrieving current stopping locations and defining individual timetable based on assigned location, Patent Numbers: WO2003005321-A;EP1402497-A;EP1274057-A1;WO2003005321-A1;EP1402497-A1;US2004135704-A1;EP1402497-B1;DE50201584-G;ES2231687-T3.

Ficher, G.,Gotthardt, K., Gunselmann, W, et al., (2005) Travel information provision system for use by public transport passengers comprises a personal device with input and communication means and a central infrastructure linked to public transport services, Patent Number: DE102004044205-A1.

Gomes Rocha, N.M; Santos, D.F.L.; Rossetti, R. J. F., (2009) myTIS: a MAS-based TIS for public transport users, Urban Transport 2009, 177-183.

Granberg, M.,(2000) Travel planner system for providing travel information and updates over a mobile network, Patent Number: EP 1091625 A2

Hamet, P., (2009) Space to grow - the progress of European Union's Galileo Global Navigation Satellite System Project, ITS International Magazine, Charging & Tolling, Jan. /Feb., 42-43.

Homburg, P.C., (2001) The Architecture of a Worldwide Distributed System, Dissertation.

Jennings, N.R., (2001) An agent-based approach for building complex software systems, Communications of the ACM, 44, 4, 35-41.

Lehmann, H., (1970) Client/Server Based Statistical Computing, Dissertation.

Malan, R., Bredemeyer, D., (2002) Software architecture: Central Concerns, Key decisions.

Mucke, R., (1988) Travel information system for public transport – uses sequentially illuminated stopping points marked along route and counter indicating next or last stop, Patent Numbers: EP258876-A2; EP258876-A; DE3629977-A; DE3629977-C2; EP258876-B1;DE3750999-G.

NIPPON DENKI TSUSHIN SYSTEM KK, (2003), Boarding point guide notification system for public transport system, transmits guide information to personal digital assistant of user, based on travel information and positional information of personal digital assistant, Patent Numbers: JP2003276607-A;JP3970650-B2.

Ort, E., (2005) Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools, Sun Microsystems.

Parsons, S., Wooldridge, M., (2000) Game Theory and Decision Theory in Multi-Agent Systems, AI Magazine, Kluwer Academic Publishers, 20, 55-68.

Russell, S., Norvig, P., (1995) Artificial Intelligence: A Modern Approach, Prentice-Hall Inc, 2, 31-51.