BIONETS

WP 2.2 – PARADIGM APPLICATIONS AND MAPPING

TI

Deliverable 2.2.3

AUTONOMIC CONTROL LOOP BASED ON SOCIAL MODELLING

Reference:	BIONETS/WP2.2
Category:	Deliverable
Author(s):	Antonietta Mannella (TI)
Editor:	Antonietta Mannella (TI)
Verification:	Paolo Dini (LSE); Daniele Miorandi (CN);Francesco De Pellegrini (CN)
Date:	27.07.2007
Status:	Final
Availability:	Public

Executive Summary

The objective of this deliverable to analyze the Social Network Analyis classical approach to find out how it can be applied to the “islands” of Bionets nodes to define management mechanisms based on social cooperation among nodes for supervision issues. This deliverable is developed within WP2.2 “Paradigm Applications and Mapping" activities where Social Network Analysis (SNA) can be used as a high-level investigation to characterize relationships and peer-to-peer interactions between U-Nodes. However, this deliverable does not claim to cover entirely all the aspects introduced; it is intended to provide a starting point for more detailed further analysis.

Document History

Version History

Version	*Status*	*Date*	*Author(s)*
0.1	*First Draft*	*05. 02.2007*	*A.Mannella(TI)*
0.2	*Draft*	*05.04.07*	*A.Mannella (TI), Paolo Dini (CN)*
0.3	*Draft*	*22.06.2007*	*Paolo Dini (LSE),Daniele Miorandi (CN)*
0.4	*Draft*	*19.07.2007*	*F.De Pellegrini (CN);Paolo Dini (CN)*
0.5	*Draft*	*25.07.2007*	*A.Mannella (TI)*
1.0	*Final*	*27.07.2007*	*A.Mannella (TI),Daniele Miorandi (CN)*

Summary of Changes

Version	*Section(s)*	*Synopsis of Change*
0.1	*Not applicable*	*Table of Contents;First Draft*
0.1	*3*	*Use case added*
0.2	*2,3*	*Table of Contents changed, added 2 and 3*
0.3	*1,2*	*Improved sections*
0.4	*3*	*Completed section*
0.5	*1,3,4*	*Modified Introduction, improved 3.4, conclusion added*
1	*all*	*Final editing*

Contents

1. Introduction 6
1.1 Motivation and Objectives 6
2. Autonomic Control Loop 7
2.1 Existing Approaches 9
2.1.1 Mape-k model 9
2.1.2 Rainbow 10
2.1.3 COLV 11
2.1.4 Kinesthetics eXtreme and Olives 12
3. Social Network Analysis and U-nodes 13
3.1 Survey of Social Network Analysis 13
3.2 Social Network Modelling in Bionets: U-nodes Ecosystem 14
3.3 Social Network and Social Network Analysis in Bionets 15
3.4 Autonomic control loop based on social network modelling between U-Nodes 16
4. Conclusion and future works 19
5. References 20
6. Terminology 21

1. Introduction

1.1 Motivation and Objectives

Social network analysis (SNA) is a branch of Network analysis and its main concern is the study of social networks to understand their structure and behaviour. The foundation [1]of the modern SNA is essentially based on the study carried out in the 1930s by Jacob Moreno in sociometry as an attempt to quantify social relationship. Also in the same years, Harvard University researchers pointed their attention on the study of cohesive subgroups (cliques) in certain social contexts such as work, family, associations. The term "social network" appeared for the first time in the 50s by a group of anthropologies in Manchester who were mostly interesting in people's informal relationships rather than in those linked with institution or associations. Further studies led researchers to introduce set theory for representing social structures and to improve methods of study and analysis used in modern SNA.

In social network theory, society is conceived and studied as a network of relations. The basic idea is that every individual (or actor) is in relation with others and this interaction moulds and modifies the behaviour of all of them. The main goal of the network analysis is to discover, analyze, and characterise the ties between actors. A Social Network (SN) is a structure, whose nodes are generally individuals, that shows connections of some kind (e.g. e-mail exchange, acquaintance etc.) between the people in that population.

The aim of this deliverable is to study the appliance of Social Network Analysis principles in the Bionets Project without giving any implementation details. In fact at this stage we are more interesting in analyzing the SNA classical approach to find out how to apply it to the "connected islands" [2] to define management mechanisms based on social cooperation between nodes. Basically this means to understand how Social Network Analysis will support the feedback phase where run-time modifications to the system are carried out automatically for optimizing the network structure or to face unexpected behaviour when occurs (e.g.,,changes in network configuration, service availability).

In Bionets we could use SNA from two different perspectives: as a high-level tool to characterize relationships and interactions between nodes and as a driver to mould the network to the services it runs.

In the first case we have to define suitable social models reflecting social aspects of U-Node cooperation to achieve a new management model and this is the main objective of this deliverable; the second issue is closer to the classical approach of social network analysis where we deal with communities of people to get in touch with new (possibly) trusted contacts that share the same services. Services can be shared both at the usage level (i.e. users can access services already deployed) and at the service definition level (i.e. services are created upon users' requests and are potentially redefined by other users). Then Social Network Analysis can be used to highlight the structure of the U-Node network (and consequently of the related T-Nodes) hosting the service.

This requires taking into account the solutions for Node Cooperation defined in T2.2.1 and the disappearing middleware solutions being studied in T1.2.4.

As the scope of this deliverable is to study how to introduce SNA to define an autonomic control loop for Bionets nodes community, section 2 provides an explanation of the autonomic concepts as well as an overview of the exiting approaches in literature. In Section 3 after a brief about the state of the art in Social network analysis, the use of Social Network Analysis in Bionets is explained.

2. Autonomic Control Loop

The progress in IT and ICT has resulted in the exponential growth of scale and complexity in computing systems, which presents obstacles to further development: configuration, healing, optimization, protection and in general management challenges are beginning to affect the capabilities of existing tools and methodologies, and are rapidly rendering the systems and applications almost unmanageable, not optimised and insecure.

IBM introduced the Autonomic Computing initiative in 2001[13], with the aim of developing self-managing systems to overcome the complexity of the actual systems. The word autonomic is inspired by the human body's Autonomic Nervous System, part of the nervous system that controls the vegetative functions of the body (e.g., circulation of the blood, heart rate, body temperature, the production of chemical 'messengers', (i.e. hormones) etc.) thus hiding to the conscious brain the burden of dealing with these and many other low-level, yet vital, functions. In a similar way, autonomic computing refers to the self-managing characteristics of computing resources as such, capable of hiding their complexity to operators and users. Systems make decisions responding to high-level policies from operators. An autonomic system will constantly check and optimize its status and automatically adapt to changing conditions.

Upon launching the Autonomic Computing initiative, IBM defined four general properties a system should have to constitute self-management: self-configuring, self-healing, self-optimising and self-protecting. Since the launch of Autonomic Computing, the self-* list of properties has grown substantially, including features such as self-anticipating, self-adapting, self-organized, self-recovery, self-reflecting, etc.

In the context of this Deliverable, the following definitions have been taken: an automatic system is capable of providing "automatic responses" to predictable events. It is a system working without (or with limited) human intervention. Normally it reacts to pre-defined stimuli with pre-defined outputs.

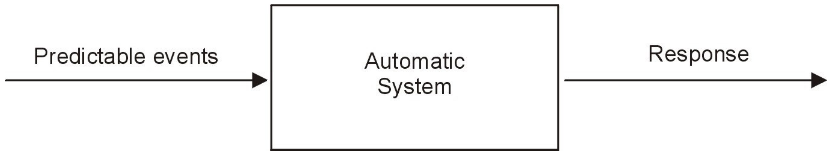


Figure 1. Automatic System

An autonomic system is capable of providing "automatic responses" to unpredictable events.

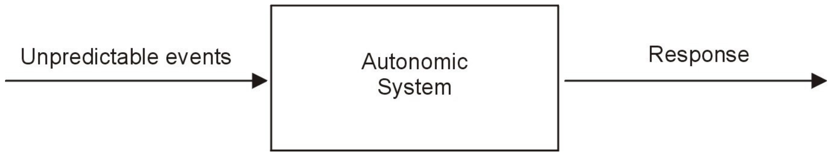


Figure 2. Autonomic System

It's a system that reacts "automatically" even to unpredicted stimuli, thus showing capabilities of adapting dynamically to variable contexts and situations ("mostly" independently of human operators). It shows "learning" and "self-adaptation" features.

In a context where a system input variable may assume an unpredictable number of configurations, it is impossible to define all possible system outputs. An autonomic system is capable of surviving in such a context as it should be able to provide "automatic responses" to unpredictable configurations (unpredictable events). As all possible configurations cannot be pre-defined, the systems itself elaborates and provides unpredictable input-output correspondences starting from some pre-established information initially provided to the system. In case a system won't be able to show this autonomic behavior, it cannot survive in (or adapt to) such a context.

An autonomic system needs to have some built-in mechanism to check its own health, which paves the way to an intelligent control loop able to collect information from the system, make decisions, and then adjust the system where required.

2.1 Existing Approaches

There have been a number of research efforts in both academia and industry to develop autonomic systems and applications. For example research in autonomic architectures consists of general architectures for individual components or complete autonomic computing systems, based on the integration of advanced technologies like Open Grid Computing, Web Services, (Multi-) Agent technologies, and/or intelligent/autonomous robotics. In the following section some of these approaches are analyzed and described.

2.1.1 Mape-k model

The IBM vision of autonomic computing implies that developing self-managing systems involves an intelligent control loop named MAPE (monitor—analyze—plan—execute) which collects information from the system (i.e., through sensors); analyzes the information to make decisions, and then adjusts the system where required through effectors.

.

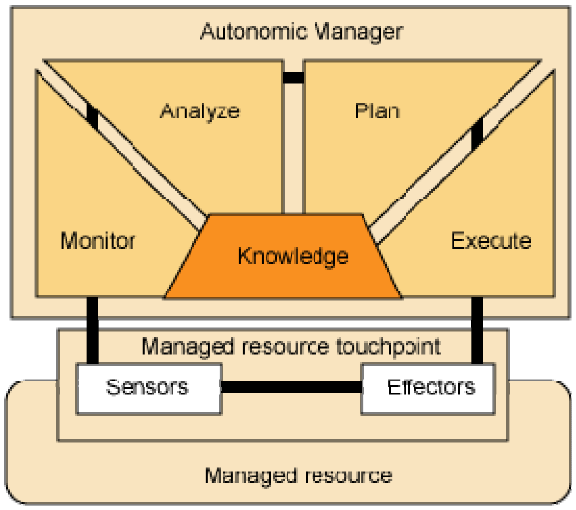


Figure 3 IBM MAPE-K model

As depicted in figure 3 according to the IBM vision, an autonomic element implements a control loop consisting of an autonomic manager that controls and represents the managed element. The managed element could be a resource (e.g., CPU, database, a directory service, web service, or a legacy system). The autonomic manager monitors the managed element and its external environment, and constructs and executes plans based on an analysis of this information.

The MAPE-K model adopts an extrinsic approach of controlling a system which is delegated to the autonomic manager external to the system under supervision.
The approach envisioned in Bionets is closer to [14], where a system checks its own health through a pervasive supervision, i.e., a type of immune system that continuously senses a system and reacts and infers if pathological behavior occurs. If we consider the MAPE-K model, the role of SNA is to build a supervision system starting from the ties of Bionets components (i.e., U-Nodes, T-Nodes) seen as a social community.

2.1.2 Rainbow

The Rainbow architecture [18] aims to achieve the storage and restoration of configuration data in distributed autonomic systems. The basic idea is to use configuration snapshots to restore the configurations of a system that have proved useful within a given usage context of the system.
Rainbow [19] is based on a so-called autonomic configuration language named Neptune and on the implementation framework Cloud.
The Neptune Scripting Language enables axioms, norms, and governance rules to be symbolized within an introspective object framework, such that the individual constructs that comprise a logical statement or assignment can be both inspected and modified without recompilation at runtime

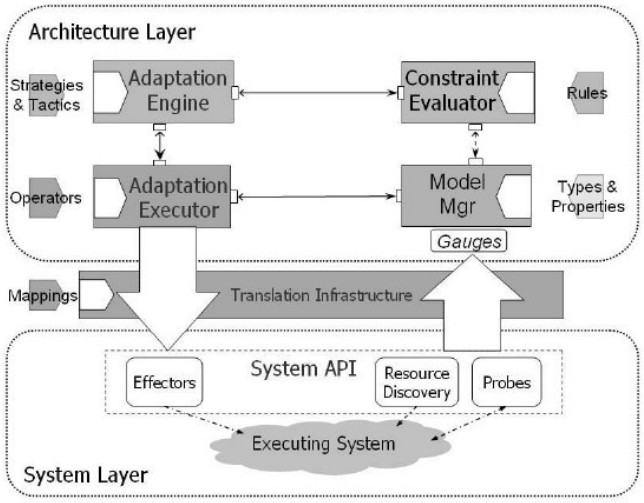


Figure 4 Rainbow Architecture

As stated in [19] "*a Cloud can be thought of as a federation of services and resources controlled by a system controller and discovered through a system space. In a distributed system, oftentimes services and dependencies can overlap with different configurations on different systems. Systems based on the Cloud framework can interact with each other, sharing and pooling resources for greater efficiency over a large deployment such as an enterprise. Neptune objects are executed on demand through an event model exposed by the Clouds architecture yielding a powerful extensible platform that is both wholly configurable at runtime, and that can be modelled at runtime*".
Clouds interact by means of a shared distributed data storage that acts as a dashboard to publish and to access information. Always referring to [19] "*at the centre of a Cloud is the System Controller (SC), a distributed service that controls access to and from the individual services and resources that are within the cloud. The SC brokers requests to services based on the system status and governance rules defined in Neptune Objects*".
The Rainbow architecture of Figure 4 comprises the following elements:

- System-layer infrastructure where the necessary system access interfaces are defined. A system measurement mechanism, realized as probes, observes and measures various states of the system. The system information may be published by or queried from the probes. A resource discovery mechanism can be queried for new resources based on resource type and other criteria. An effector mechanism carries out the actual system modification.
- Architecture-layer infrastructure gauges aggregate information from the probes and updates the appropriate properties in the model. A model manager manages and provides access to the architectural model of the system. A constraint evaluator checks the model periodically and triggers adaptation if a constraint violation occurs.

An adaptation engine determines the course of action and carries out the necessary adaptation.

Rainbow provides an extrinsic additional control layer responsible for monitoring and context dependent restoration of system configurations. An explicit system model is used to interpret and to classify monitored information on the current system state and its configuration. These contexts should not be confused with situational information, as in the previous approaches, since in this case an explicit notion of the environment of a system is not given.

2.1.3 COLV

Continuous On-line validation (COLV)[20] involves the monitoring and adaptive control of already deployed services by exerting reactive and proactive control measures that keep the running service in check with respect to its nominal functionality. The aforementioned measures thus implement active validation policies, by dynamically adapting the running service, e.g. by modifying its configuration, or its operating parameters, or by installing functional patches on the fly, etc. The range of dynamic adaptation policies that can be applied to a service may be seen as a derivation of service specifications, and each dynamic adaptation action that is taken is in itself a subject of on-line validation from that moment on.

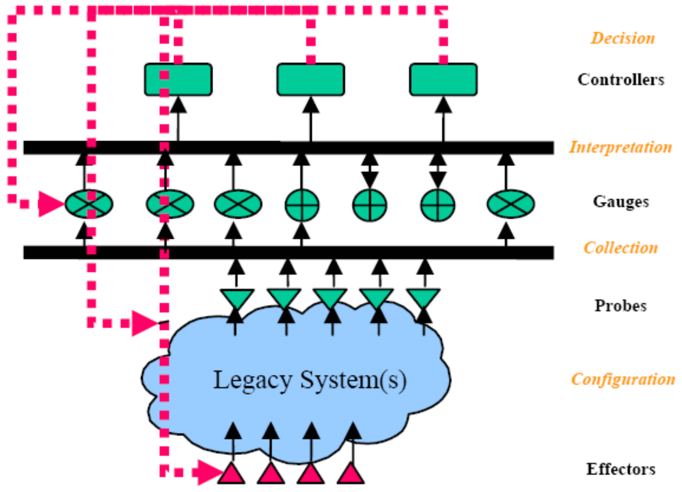


Figure 5 COLV Architecture

The control loop variant of the COLV approach is shown in 5.

- Probes are generally small, constrained, non-invasive pieces of code which get installed in or around the target application system—they may inject source code, modify byte codes or binaries, replace DLLs or other dynamic libraries, inspect network traffic, and/or perform other related tasks to collect this information;
- Gauges are responsible for interpreting data from these probes, and generate semantic events about the behaviour of the application—often operating in an effective hierarchy where higher-level gauges interpret aggregate events from lower-level gauges;
- Controllers receive analysis results from the gauges, and decide if and when to coordinate one or more effectors to attempt a repair.
- Effectors apply reconfiguration or repair, usually tuning or replacing an individual component, or spinning up a new component, as per the task(s) defined by relevant controllers.

2.1.4 Kinesthetics eXtreme and Olives

The Kinesthetics eXtreme (KX) [21] approach uses a very specific effector concept that is based on mobile code, called Workflakes. According to a task decomposition strategy, a Workflakes process generates, configures, activates groups of effectors, and coordinates them towards actuating the desired side effects onto the running controlled system. Worklets are code carrying agents that Workflakes selects as effectors, configures and dispatches onto the target system, as a side effect of process steps. Each Worklet carries Java mobile code snippets, and deposits them onto one or more target components, according to a programmable trajectory. Once deposited, the execution of Worklet code is governed by constructs that specify conditional execution, repetition, timing, priority, etc. The agent transport facilities and the code execution environment are provided by a Worklet Virtual Machines residing at all "stops" in a Worklet trajectory.

The KX approach has been continued in the Olives project [22] and further it was developed in [23] under the main keyword “system auditing”. What is novel in this work is the employment of testing technology for the interpretation of monitored data and the classification of fault states, namely the test system definition language TTCN-3 [23] . TTCN-3 has been developed in the context of protocol performance testing and it combined the classical notion of decision trees with interaction triggered branching with procedural elements for computations and data analysis. In particular, it defines means for the set-up of distributed dynamic configurations for data correlation systems and thus provides means for self-adaptation of the auditing system.

3. Social Network Analysis and U-nodes

3.1 Survey of Social Network Analysis

A Social Network is a structure, whose nodes are generally individuals, that shows connections of some kind (e.g. e-mail exchange, acquaintance etc.) between the people in that population. Social Network theory views social relationships in terms of *nodes* and *ties*. Nodes are the individual actors within the networks, and ties are the relationships between the actors. There can be many kinds of ties between the nodes: links can be strong or weak according to the number of connections between nodes. The analysis of the flows between nodes characterizes nodes as active, passive, permanent, and transient. In its most simple form, a Social Network is a map of all of the relevant ties between the nodes being studied. The network can also be used to determine the social capital of individual actors, i.e. defined as the utility associated with a person's location in a structure of relationships. These concepts are often displayed in a Social Network diagram, where nodes are the points and ties are the lines.

Social network analysts use the simple concepts of nodes, ties and flows, to derive relational matrices for everything for which connections exist. They use two kinds of mathematical tools to represent information about patterns of ties between social actors: graphs where vertices represent entities and edges represent their relationships, and matrices where relations between entities are labeled 0 or 1.

To understand networks and their participants, Social Network Analysts evaluate the *location of actors in the network*.[8]. Measuring the network location is finding the *centrality* of a node. These measures give insight into the various roles and groupings in a network -- who are the connectors, experts, leaders, bridges, isolated nodes, where are the clusters and who is in them, who is at the core of the network, and who is at the periphery. The three most popular centrality measures are: degree, betweenness and closeness.

Degree centrality is the number of direct connections a node has. Social network researchers measure network activity for a node by using the concept of degrees, which is useful to find out the hub node. In betweenness or centrality, a node is directly connected only to those other nodes that are not directly connected to each other. It is a centrality measure of a vertex within a graph. The closeness centrality deals with the mean geodesic (i.e., the shortest path) between a vertex and all other vertices reachable from it**.** Moreover the closeness nodes are in an excellent position to monitor the information flow in the network; they have the best visibility into what is happening in the network.

Each of the three approaches (degree, betweenness, closeness) describes not only the locations of individuals in terms of how close they are to the "centre" of the action in a network but they are the kind of data for which SNA is most appropriate. For example closeness can be regarded as a measure of how long it will take information to spread from a given vertex to others in the network while a node with high betweenness has great influence over what flows, and what does not, in the network but at the same time it could be a possible single point of failure for the network.

Although Social network theory was born as a branch of social science, it has been widely used and explored in different fields that are the basis of social networking sites e.g., LinkedIn [3], Orkut [4]: virtual online communities where people can get in touch with new trusted contacts that share their interests (e.g. job offers). These networks add a component of “trust” to contacts that are suggested to individuals, because such contacts can be reached through a chain of “friend of friend” steps, i.e., through a chain of mutually trusted people. Another way to exploit social network analysis is to highlight the distribution of people’s interests in a community and to see how they are related to people’s social relationships.

Social Network mechanisms are widely used to enforce cooperation in P2P systems especially to avoid selfish behavior among the peers. In [5] the authors describe an example of how to discourage selfish behavior and encourage altruistic behavior in a file-sharing application using an algorithm based on social behavior. Social Networks could be a useful methodology to optimize the distribution of peers' workloads over the overlay network. Tribler [6] is an example of social based P2P file sharing which facilitates the formation and maintenance of social networks, and exploits their social phenomena for improved content discovery, recommendation, and sharing. Tribler is an extension of BitTorrent [7], which has been one of the most popular P2P file sharing system for several years and it, according to the authors, is able to address four important challenges:

- Decentralization of the functionality of a P2P system across the various peers using Social groups which form a natural method to efficiently decentralize P2P systems as communication is mostly localized amongst group members.

- To guarantee the availability of a P2P system as a whole using proven social incentives such as awards and social recognition could stimulate users to leave their P2P software running for longer periods, thus improving the overall availability of the network.

- Integrity of the system and to achieve trust amongst peers can be solved with a social-based network, in which users actively help to clean polluted data and users can select trustworthy representatives

- Network transparency by solving the problems caused by dynamic IP addresses, NAT boxes, and firewalls. Social networks enable communicating peers to automatically select trusted mediators from the members of their social proximity, who are still online; hence, the need for fixed mediators is eliminated.

3.2 Social Network Modelling in Bionets: U-nodes Ecosystem

An overall picture of the implications of applying SN in Bionets could be explained in Figure 6 where the p2p music application *undersound* [12] is used as a case study. In the *undersound* scenario people in the Underground can upload and download, with their mobile devices (U-nodes), Creative Commons-licensed songs from local repositories (T-nodes) located at the stations, and then exchange these songs with each other during their journey. *Undersound* will also allow people to send messages to each other and to become aware of music interests or friends they might have in common. One of the interesting aspects of this scenario is to consider Social Networks from two different perspectives: the "community" of users using the same service (*undersound*) and the community of U-Nodes able to provide the service.

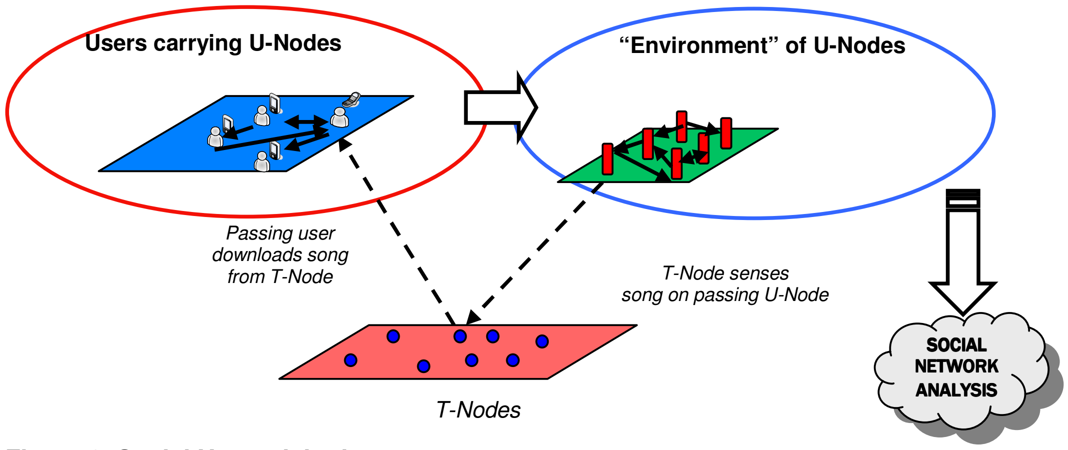


Figure 6 Social Network in the *undersound* scenario

Both communities, users using the service and the U-Nodes/T-Nodes providing it, could be subjects for the application of Social Network Analysis in such a pervasive environment where the U-Nodes are part of an ecosystem. Therefore they establish interconnected communities of living organisms in which they manage interaction with each other and with the environment that they inhabit to ensure the delivery of the specific service they host.

Hence one of the main issues of applying Social Networks in Bionets is to define appropriate models that will capture social aspects in the cooperation of U-Nodes and that will lead to the feedback phase, where run-time modifications to the system are carried out automatically when critical differences between the runtime behavior of the system and the expected one are detected. This could be achieved by inferring suitable social models reflecting social aspects of U-NODE cooperation to obtain a new management model. The results in this case are threefold:

[1] To discover the network structure characterized by a two-tier communication network where nodes form connected islands of U-NODES and T-NODES and in which network operations will be driven by the services, providing an "ad hoc" support when and where needed to fulfill users' requests. In this case an interaction with the disappearing network middleware carried by Task 1.2.4 is envisioned.

[2] To analyze the network structure according to SNA parameters to find out the social network model the network is built on. SNA will be based on the collection of the run-time data provided by U-Nodes. For example, in this context the metrics of Social Network such as the concept of degrees is useful to find out the hub node, the possible point of failure for the network, etc.

[3] Dynamic adaptation: seen as an emergent behavior where the group ability is greater than the sum of each individual acting alone. Even if the capability of each node itself may be very simple, the collective behaviors and the overall functionality arising from their interactions exceed the capacities of any individual. These higher level processes can be adaptive, scalable and robust to changes in their environment.

Starting from the analysis of the feedback provided by applying SNA principles on interpretation monitored data provided by U-Nodes we could find out how networks form, grow, evolve, and change.

3.3 Social Network and Social Network Analysis in Bionets

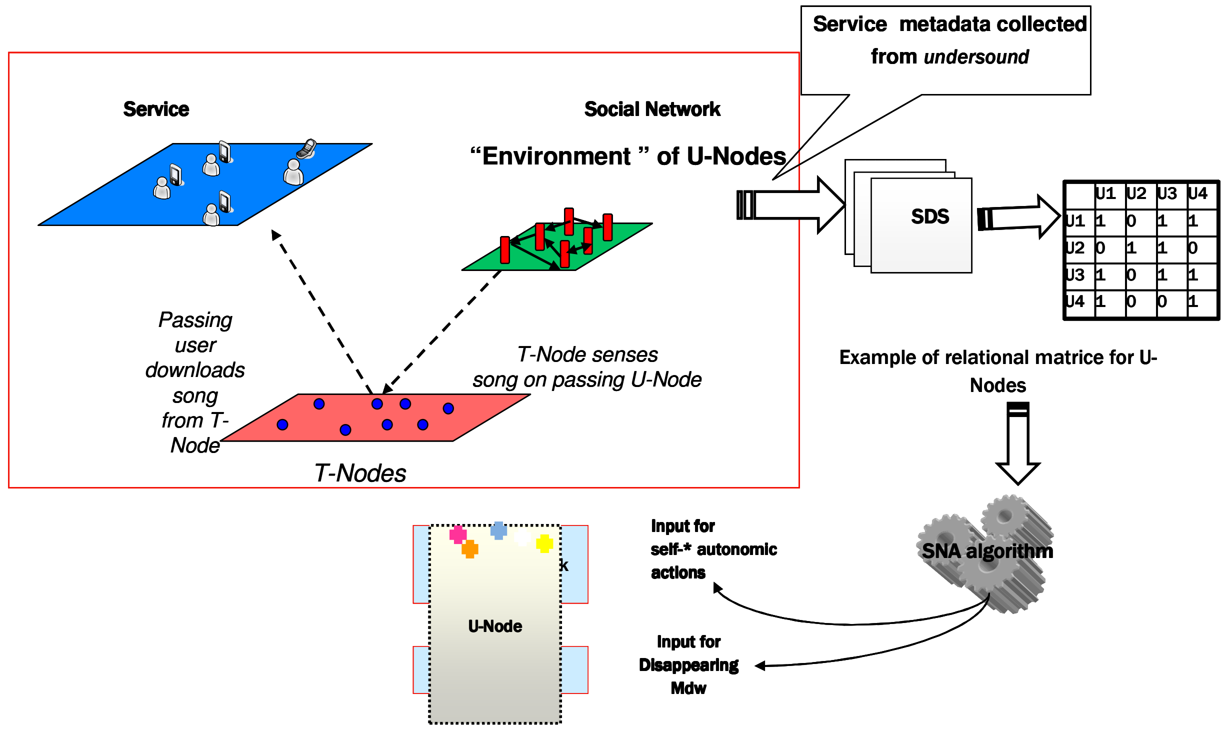
Applying SNA in Bionets should lead to an innovative and autonomic management system of the network overlay formed by U-Nodes and T-Nodes providing a particular service. The starting point of how applying SN is to analyze the cooperation between nodes; as in Bionets direct communications between T-Nodes are not allowed and that T-Nodes may communicate only with U-Nodes in proximity, then it's clear why for Social Network Analysis we concentrate especially on the communication paradigm between U-Nodes. Moreover T-Nodes, as tiny devices, are supposed to be limited in their processing capabilities and unable to perform any task to face any problem that might arise. For this reason all the autonomic features are supposed to be addressed only by U-Nodes [15] and as a consequence the SNA outcomes involve basically the U-Nodes.

Obviously the result we envisage should be flexible and scalable to apply the same mechanism to "environment of U-Nodes" belonging to different islands as a cascaded effect: as U-nodes follow the motion of their carriers (i.e., mobile users), when two U-Nodes get in proximity and begin to communicate they are not part of two different islands but we could consider them part of the same environment. Then the result of this interaction is a sort of "connection" of disconnected islands or, according to the SNA definition, the network is formed by cliques. Hence from a Bionets perspective it could be interesting to identify through SNA how network structures evolved by the aggregation of smaller ones and viceversa: cliques, n-cliques etc. A clique is a sub-set of a network in which the nodes are more closely and intensely tied to one another than they are to other members of the network.

The Bionets islands are very similar to the clique definition, which could be extended to N-clique, where N stands for the length of the path allowed to make a connection to other members in a sort of friend-of-friend chain.

3.4 Autonomic control loop based on social network modelling between U-Nodes

A model for control loop is mainly based on collecting and analyzing variables containing values of interest. The values of such variables together with constraints and relationships among them provide possible states interesting from the supervision point of view, and they allow defining proper actions in some predefined conditions. The central issue is to establish a management model for allowing the system to discover by itself the best solution dynamically, in an autonomic way. We illustrate this approach with the help of the *undersound.*
In this scenario, the technical aspects of the interaction between nodes are in terms of resource sharing and reliable transmission; then if we consider this use case from a management point of view, we could analyze the node cooperation based on the information (service metadata) nodes exchange with each other through the use of messages, which is one of the interaction model implementation considered in the Bionets Service Architecture [15].As described in [12] T-Nodes are located in fixed positions in the area understudy/monitoring and collect context information from the surrounding environment. When in proximity, the upper-layer U-Nodes gather Context information from the T-nodes. U-Nodes are user portable devices such as PDAs, laptops or smartphones. They are more complex and powerful devices than T-Nodes and are carried by people in their everyday life. In the *undersound* communications need to rely on contact opportunities that may arise during the motion of U-nodes. Whenever two U-nodes are in the communication range of each other they can exchange the information they carry. This information is described in terms of service metadata collected when the user is engaged in some transaction: uploading songs, downloading songs or scanning for other U-Nodes in zone.
From MAPE-K perspective the metadata collected constitutes the knowledge used by the analyzer to provide awareness of the external environment; while the planner and executor decide on the necessary self-management behavior that will be executed through the effectors. The Social Network Analysis used in this context is halfway between the analyzer and the planner: starting from the opportunistic metadata exchanges, through the SNA techniques we could analyze the network activity and as a consequence to plan some corrective actions when necessary.
Figure 7 shows a logical overall picture of what we envisage as autonomic control loop based on social network analysis taking into account also some of the Bionets Service Architecture elements [15].


Service metadata collected from *undersound*
Service
Social Network
"Environment " of U-Nodes
SDS
Passing user downloads song from T-Node
T-Node senses song on passing U-Node
T-Nodes
Example of relational matrice for U-Nodes
SNA algorithm
Input for self-* autonomic actions
Input for Disappearing Mdw
U-Node

	U1	U2	U3	U4
U1	1	0	1	1
U2	0	1	1	0
U3	1	0	1	1
U4	1	0	0	1

Figure 7 SNA applied in Bionets in the *undersound* scenario

As the *undersound* should work on top of the Bionets Service Framework (BSF) using the facilities it provides, it makes sense to consider the service metadata into the Shared Data Space (SDS) used by BSF for content delivery, service control, message management. Furthermore in our scenario the service metadata is collected when the user is engaged in five types of transactions and then we suppose that each of them produces a message which carries its own specific metadata described in [12]. For sake's clarity we assume that messages have the same name of the associated transactions listed below

[1] Uploading to a station access point
[2] Downloading from a station access point
[3] Downloading from another user
[4] Uploading to another user
[5] Scanning for users or tracks in range

The metadata carried by the aforementioned messages are in charge of the Bionets Service Framework and collected during the node cooperation in Shared Data Space. The BSF should provide the techniques to organize suitable views of the information collected in terms of Social Network Data (i.e., the actors and the ties) necessary for Social Network Analysis. The Social Network Data is usually represented by a correspondence matrix M where:

$M(i,j) = 1$ denotes the existence of a link between U-Node i and U-Node j
$M(i,j) = 0$ the absence of such direct connection

This correspondence matrix between nodes is generally used by the SNA algorithms (figure 7) to measure network activity for a node and to forecast a possible single point of failure for the network or whatever parameters useful to measure the node "fitness" .The fitness concept is ongoing to be defined in Bionets [16] although for the control loop purpose the fitness describes the difference between the expected and the observed behaviour/performance of a U-node.
The fitness could be considered as an endogenous variable that helps to measure functional parameters both for the U-Node and for the island it belongs to; referring to the *undersound*

scenario a functional parameter for a single node could be the number of songs successful saved on a user's device while from the island perspective this means to measure network bandwidth consumed in order to download a song. Then the usage of SNA in a control loop means measuring the "social" activities of the U-Nodes using the result (i.e., metrics) to infer corrective countermeasures at two different levels (figure 7):

- the node itself triggering the autonomic features
- the overlay structure of the network middleware.

The first aspect is strictly connected with the work ongoing on the Service architecture addressed by [15] where one of the desired features is the ability to dynamically change and adapt service's internal behavior according to any modification in the environment. The situation when a node fails and "disappears" (and implicitly the services) corresponds to loose
The second aspect is more related with the work carried out by the WP1.2 [12] where the input from SNA could help to optimize the overlay network. The degree centrality, i.e. the number of ties a node has, is useful to identify the nodes at the centre of the network (i.e. Super Peer).
At the same time the Closeness centrality approaches emphasize the distance of a node to all others and it's useful to find out who are in an excellent position to access the information flow in the network.
Another metrics used in SNA, the betweenness, is useful to establish the node with a 'broker' role in the network. In this case, it is necessary to have a complete picture of relations among nodes which also helps to properly define and measure many of the structural concepts of network analysis e.g., the density of the network. The density is defined as the sum of the ties divided by the number of possible ties; this measure may give us the speed at which information diffuses among the nodes.
At the same time the Closeness centrality approaches emphasize the distance of an actor to all others and it's useful to find out who are in an excellent position to access the information flow in the network.
Most of the commercial or freely Social Network Software analysis like InFlow [9] , UCINET [10] or Pajek [11] consists of a graph drawing tool and a certain number of network analytic routines to calculate (simple) network statistics (e.g., centrality) or more complex (iterative) algorithms (e.g., cluster analysis).
They are also able to produce data that can be used as input for other procedures, for example data sets from UCINET could be exported to Pajek. All these tools are suitable for centralized elaboration starting from the raw data matrix oriented, that is, data required for analysis are collections of one or more matrices. They are not directly useful for the high distributed Bionets environment although they could be a good starting point for the implementation of Social Network Analysis in a de-centralized environment.

4. Conclusion and future works

This deliverable describes a preliminary introduction on the use of Social Analysis methodology in Bionets project. The result of the SNA appliance should be used by the self*autonomic features of U-Nodes to infer the appropriate reaction; it should also be used by the Network Middleware under studying in Bionets project to maintain the structure of the overlay.

As SNA is strictly related to these two aspects, future work could envision a strict collaboration with these two activities taking into account the possibility to include SNA as a “service” of the Bionets platform. Furthermore another activity could be to apply SNA as a driver to mould the network to the services it runs, and services, in turn, become a mirror image of the social networks of users they serve. This should address explicitly the interaction of the user with the communication system at the service level: the change in the network paradigm will arise directly from the user behavior.

5. References

[1] John Scott *Social Network Analysis.* Newbury Park CA, 1992 Sage Editor
[2] Carreras, I., Chlamtac, I., De Pellegrini, F., Mionardi, D.: "BIONETS: Bio-Inspired Networking for Pervasive Communication Environments", 2005, available online: www.bionets.org**.**
[3] Linkedln http://www.linkedin.com/
[4] Orkut http://www.orkut.com
[5] Hales D., Edmonds B. : "Applying a Socially Inspired Technique (Tags) to Improve Cooperation in P2P Networks" IEEE Transaction on systems, man , and cybernetics —PART A: SYSTEMS AND HUMANS, VOL. 35, NO. 3, MAY 2005
[6] http://www.tribler.org
[7] http://www.bittorent.org/
[8] http://www.orgnet.com/sna.html
[9] InFlow web site http://www.orgnet.com/inflow3.html
[10] UCINET web site http://www.analytictech.com/ucinet/ucinet.htm
[11] PAJEK Web Site http://vlado.fmf.uni-lj.si/pub/networks/pajek/
[12] BIONETS Deliverable D2.2.1
[13] Horn, P.: "Autonomic Computing: IBM's Perspective on the State of Information Technology", IBM T.J. Watson Labs, New York 2001, available online: www.research.ibm.com/autonomic/manifesto/autonomic_computing.pdf
[14] Baresi, L.; Baumgarten, M.; Mulvenna, M.; Nugent, C.; Curran, K.; Deussen, P.H "Towards Pervasive Supervision for Autonomic Systems" Distributed Intelligent Systems: Collective Intelligence and Its Applications, 2006. DIS 2006. IEEE Workshop on Volume , Issue , 15-16 June 2006 Page(s): 365 – 370
[15] Bionets Deliverable D3.1.2
[16] Bionets Deliverable D3.2.2
[17] International Network for Social Network Analysis http://www.insna.org/
[18] Garlan, D., S. Cheng, A. Huang, B. Schmerl, P. Steenkiste, "Rainbow: Architecture-based Self-adaptation with Reusable Infrastructure", IEEE Computer, 37(10):46-54, Oct. 2004.
[19] Miseldine, P., A. Taleb-Bendiab, "Rainbow: An Approach to Facilitate Restorative Functionality within Distributed Autonomic Systems", PGNet 2005, Liverpool, 2005
[20] Deussen, P.H., G. Valetto, G. Din, T. Kivimaki, S. Heikkinen, and A. Rocha, Continuous On-Line Validation for Optimized Service Management, in Proceedings of EURESCOM Summit 2002, Heidelberg, Germany, October 21-24,
[21] Kaiser, G., J. Parekh, P. Gross, G. Valetto, "Kinesthetics eXtreme: An External Infrastructure for Monitoring Distributed Legacy Systems." Autonomic Computing Workshop -- IEEE Fifth Annual International Active Middleware Workshop, Seattle, USA, June 2003. http://www.cs.columbia.edu/techreports/cucs-019-03.pdf
[22] OLIVES Home Page: www.eurescom.de/public/projects/P1100-series/P1108.
[23] Deussen, P. H., G. Din, I. Schieferdecker: A TTCN-3 Based Online Test and Validation Platform for Internet Services. ISADS 2003: 177-184
[24] ETSI European Standard (ES) 201 873-1 V2.2.1 (2003-02 Methods for Testing and Specification (MTS); The Testing and Test Control Notation version 3; Part 1: TTCN-3 Core Language
[25] Knight, J. C., D. Heimbigner, A. Wolf, A. Carzaniga, J. Hill, P. Devanbu, M. Gertz, The Willow architecture: comprehensive survivability for large-scale distributed applications, Intrusion Tolerance Workshop, The International Conference on Dependable Systems and Networks, Washington, DC, June 2002
[26] Carzaniga, A., D.S. Rosenblum, and A.L. Wolf, Design and Evaluation of a Wide-Area Event Notification Service, ACM Transactions on Computer Systems, 19(3):332-383, Aug 2001.
[27] Hanneman, Robert A. and Mark Riddle. 2005. Introduction to social network methods. Riverside, CA: University of California, Riverside (published in digital form at http://faculty.ucr.edu/~hanneman/)

6. Terminology

AP-node	Access point to existing network infrastructure
BIONETS	BIOlogically-inspired autonomic NETworks and Services
Service	A BIONETS architecture entity offering services to users and other services
SNA	Social Network Analysis
T-node	Simple sensor nodes
U-node	Complex nodes offering services to users