**SEVENTH FRAMEWORK PROGRAMME**
**THEME 3**
**Information and Communication Technologies**

**Grant agreement for:**
Collaborative project, Small and medium-scale focused research project (STREP)

Deliverable D4.5:

# Second Report on the Unification of Results

**Project acronym:** FRONTS
**Project full title:** Foundations of Adaptive Networked Societies of Tiny Artefacts
**Grant agreement no.:** 215270

**Responsible Partner:** RACTI
**Report Preparation Date:** Friday, 12 February, 2010

# Contents

# 1 Introduction

The aim of the FRONTS project is to establish the foundations of adaptive networked societies of small or tiny heterogeneous artefacts. In a nutshell, we are working towards a science of adaptive organization of large networks of small or tiny artefacts.

Our approach to this goal is foundational and is based on algorithmic expertise. We intend to apply our models, methods and results to the scrutiny of large-scale simulations and experiments, from which we expect to obtain valuable feedback.

Our methodology follows basic principles of System Science. The networks of our study are (rather complex) "systems" of small and restricted parts, that may be organized via local communication. Dynamicity (and change) is a main characteristic, both for the network and also of its environment. Any computation is locally restricted, and no central control exists. Thus, the nets (systems) considered are *distributed* and cooperation among the tiny parts is not automatically guaranteed. Rather, it *has to be established* (in ad-hoc situations) and *maintained*, while adapting to dynamically changing external situations.

Still, if such systems are to be designed, they should be *"programmable"* and *trusted*. FRONTS aims to deliver at its end a set of well defined design rules for such systems.

We have decomposed our effort into four technical workpackages. As the contract indicates, they do no step in isolation, but they interact in a designed way. Here, we rephrase their goals:

- **WP1:** Aims in stating Models, Laws and Complexity measures for our networks.
- **WP2:** Aims in answering how the "net" prepares internally in order to be "ready to adapt".
- **WP3:** Aims in answering how the "net" reacts successfully to a dynamic external environment (how it adapts).
- **WP4:** Aims in verifying our methods and models via experiments and simulations.

For the 2nd year, we promised and delivered 5 technical deliverables, for the 4 WPs and the unification of results:

**D1.2 (WP1):** Models and Algorithmic Consequences for Dynamicity, Restricted Locally Computation, and Cooperation

**D2.2 (WP2):** Set of Schemes for Designing the Adaptable Network Infrastructure

**D3.2 (WP3):** Set of Schemes for Adapting to the Dynamic Environment

**D4.4 (WP4):** A First Set of Well-Designed Simulations, Experiments and Possible Benchmarks

**D4.5:** Second Report on Unification of Results

The effort of unifying our results, in the 2nd year of FRONTS, was seriously affected by the recommendations made by the review process at the end of the 1st year. We summarize here the main points of the recommendations:

**R1:** Concentrate on unification

**R2:** Focus on experimental work

**R3:** Consider innovative network technologies, e.g., opportunistic networks

**R4:** Communication graphs is not a good idea (wireless link is an ill-defined notion)

**R5:** Not only Shawn, not only simulations

The project took the recommendations into a very serious consideration. We provide below a high-level view of our answers and actions resulting from this process. We judge that the recommendations were of a great help in the direction of focusing our results and unifying our actions.

## Highlight of Actions/Answers to the Recommendations:

**R1:** We constructed and studied 2 unifying scenaria:

- **1.1.** Multi Radio Pedestrian Energy Scavenging Sensor Nets (MURPESS) for WP2 (5 partners involved, FRONTS-TR-2010-7)
- **1.2.** Tracking/monitoring wildlife (e.g., a society of lizards) for WP3 (5 partners involved, FRONTS-TR-2010-11)

We also designed and performed *7 unifying experiments* (all in our Central Experiments Repository), in each of which a subgroup of partners contributed.

**R2:** We did 7 unifying experiments and 9 stand-alone experiments, resulting in 25 Software Components, all implemented and tested.

**R3:** We considered opportunistic networks (in D2.2) and designed novel, fully decentralized recommendation techniques.

**R4:** We focused mainly on modeling communication as local interactions (rendezvous of devices), mostly uncontrolled by the protocols. We also focused on dynamic (changing due to mobility) communication settings.

**R5:** We did our experiments on a variety of platforms, including hardware platforms, even a HW platform for pervasive games which gained golden awards (FRONTS-TR-2009-31: Developing Multiplayer Pervasive Games and Networked Interactive Installations using Ad hoc Mobile Sensor Nets, in the *5th International Conference on Advances in Computer Entertainment Technology (ACE 2009)*.

Our *unifying* experiments were designed in advance by the Consortium and met the following criteria:

(1) To be of joint interest to at least 3 partners;

(2) To be jointly implemented by these partners;

(3) To focus on crucial issues of the WPs.

# 2 The decisive steps (and successes) per WP: A quick overview

## 2.1 WP1: Principles of Adaptively Organized Societies of Artefacts

We have advanced and focused the models initially considered in the first year. Figure 1 is a high-level picture of the current status:

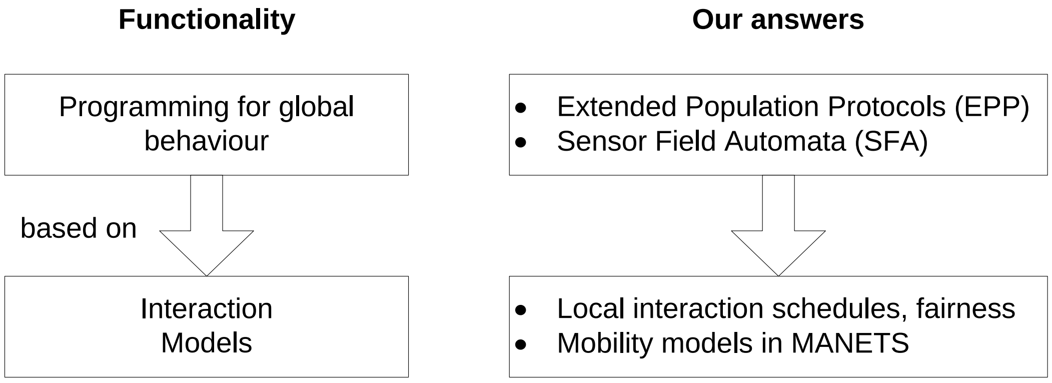
Functionality
Our answers
Programming for global behaviour
based on
Interaction Models
• Extended Population Protocols (EPP)
• Sensor Field Automata (SFA)
• Local interaction schedules, fairness
• Mobility models in MANETS

Figure 1: Focusing the models

**Models for asynchronous local interactions.**

Our Extended Population Protocols (EPP) include two basic submodels, both dealing with the use of memory into the (basic) Population Protocols (PP): Our Mediated

Population Protocols (MPP) assume a global passive storage of one state per interaction pair. We have shown that they are stronger than the PP. Our recently developed PALOMA model (Passively Mobile Log-Space Machines) assume that each device is now having a local memory of size logarithmic to the order of population size. E.g., for $n \leq 2^{100}$ agents we need 100 local bits.

In both MPP and PALOMA, protocols are just a set of local interaction rules for device pairs. In MPP the rules involve also the state of the interaction pair. In PALOMA, the devices can compute between interactions. PALOMA is also stronger than PP, and seems more easily *implementable*. We have shown very recently that PALOMA can compute all predicates that a Turing Machine of SPACE($n \log n$) can do!

Both models are *scalable* (same rules for any population size) and make no assumptions on existing id's of devices (*uniform*). *There is no notion of a "link" or communication graph.* Instead, interactions of devices are scheduled by an adverse (but fair) scheduler.

**Models for continuous interaction with the environment.**

We have developed the Sensor Field Automata (SFA) model. This is a synchronous model (works in rounds) and devices get data from neighbours *and also from the environment in each round.* Then, devices locally "compute" and send (broadcast) data to any neighbours *and the environment*, *at end of each round*.

Thus, SFA computes output *streams*, provided some input streams of data. We have shown that, under some assumptions about streams, SFA can simulate the basic PP model.

**Remarks for EPP and SFA.**

FRONTS judges that both models capture pervasive and adaptive societies of tiny artefacts and that they are complementary. Both models are inherently adaptive (may admit various stable "outputs" depending on input and rules). Both EPP and SFA are *programmable* (e.g., SFA by the Language "Nets in Motion") and *verifiable* (e.g., by our unifying experiments). In our experiments we show that EPP can program real sensor networks.

**Dynamicity (Mobility) Models.**

In such models, nodes "travel" according to some stochastic motion and interact when they are close to each other. They capture interaction schedules in highly mobile and delay-tolerant networks and can be modeled by Dynamic Geometric Graphs.

They allow for studying information spread and our project has discovered a nice connection to thermodynamics!

## 2.2 WP2: Designing the Adaptable Network Infrastructure

Already in the contract of FRONTS we had distinguished three parts of the internal organization activity:

**Task 1:** How to re-organize the *Communication Infrastructure*?

**Task 2:** What are the *roles of artefacts* that can be re-adjusted? What are the methods to do this?

**Task 3:** What is the approach to re-organize the *security* of the net?

For each part, FRONTS came out by proposing some basic *functionalities*. They are listed in Table 1.

Table 1: Continuous Re-organization

| Part of the System | Functionalities needed |
|---|---|
| Internal Communication | ▷ Reorganize *data gathering*<br>▷ Allow for *redundancy in connectivity*<br>▷ Maintain *hierarchy* in the communication graph that can keep a consistent status of the network |
| Redefine Roles of Artefacts for internal Reorganization | ▷ Local Learning Methods<br>▷ Trust enhancement<br>▷ "Equilibria" and economic approaches for self-organization<br>▷ (Node "colors" are important to indicate and code local constraints) |
| Security | ▷ Privacy Protection (keys and identities management)<br>▷ How to secure the routing? |

In Year 1 we have identified some initial approaches to the required functionalities.

### The work in Year 2.

The consortium first worked out a unifying use case scenario: Multi Radio Pedestrian Energy Scavenging Sensors (MURPESS) (FRONTS-TR-2010-7).

We then proceeded to present *five schemes* that meet the needed functionalities foreseen in the 1st year. Our schemes are very innovative and they consider also mobile ad-hoc nets with only sporadic communication (e.g., Scheme 3). Our schemes, their benefits and their *provided* functionalities are shown in Tables 2 and 3.

Table 2: Schemes and their benefits

| | Scheme | Benefits |
|---|---|---|
| 1 | Virtual coordinates for "geographic" routing | ▷ avoids dead-end zones<br>▷ energy efficient |
| 2 | Dynamic allocation of files in mobile ad-hoc nets | ▷ online<br>▷ adheres to changes in the net |
| 3 | Smart tags and recommendation in opportunistic nets | ▷ simplicity<br>▷ fully decentralized<br>▷ sporadic communication nicely exploited |
| 4 | Distributed protection from subsystem takeovers | ▷ self-stabilizing<br>▷ dynamic cooperation<br>▷ enforcement via collective punishments |
| 5 | Security helpers via randomization of RFID tags | ▷ devices that help tiny artefacts to perform crypto operations |

Table 3: Schemes and their functionalities

| Sche | Benefits |
|---|---|
| 1 | ▷ allows for redundancy in connectivity<br>▷ can implement hierarchies |
| 2 | ▷ recognizes data gathering<br>▷ allows for redundancy |
| 3 | ▷ gives roles to artefacts<br>▷ exhibits local learning<br>good for sporadic nets |
| 4 | ▷ uses equilibria and economics approaches for self-organization<br>▷ enhances trust |
| 5 | ▷ protects privacy and secures the net<br>▷ gives roles to some artefacts |

## 2.3 WP3: Adapting to the Dynamic Environment

### Connection to the results of the 1st year.

In Year 1 we spotted 5 needed functionalities of the net, each *reacting* to some set of external challenges. We present those functionalities in Table 4 (please refer to Tables 4,5 and 6 of D4.3).

Table 4: Functionalities of the net

| | Functionality |
|---|---|
| 1 | Distributed Cooperation |
| 2 | Tracking of Resources |
| 3 | React to Physical Changes |
| 4 | Trust |
| 5 | Reliable Communication |

**The work in Year 2.**

FRONTS created a unifying scenario: How to adapt a network of sensors in order to monitor wildlife (e.g., a community of lizards) on a remote island (FRONTS-TR-2010-11).

This scenario involves tracking, creating/maintaining a communication topology in an unknown place, distributed cooperation, energy handling, avoidance of obstacles, etc.

Based on this, and the functionalities identified, we have delivered 9 schemes. The schemes and the functionalities implemented are shown in Tables 5 and 6. Our schemes are innovative and can be directly applied.

Table 5: Schemes

| | Scheme |
|---|---|
| 1 | Self-synchronized duty-cycling and minimum energy multicasting |
| 2 | Scheduling selfish distributed computation |
| 3 | Monitoring objects and disseminating the gained data in the network |
| 4 | Global network structure |
| 5 | Reacting to dynamically changing environments |
| 6 | Assigning roles within the group of sensor nodes |
| 7 | Sybil attacks |
| 8 | Energy-aware transmission policies |
| 9 | Reliable communication |

Table 6: Schemes functionalities

| Scheme No | Functionality |
|---|---|
| 1, 2, 8 | Distributed Cooperation |
| 3, 4 | Tracking of Resources |
| 5, 6 | React to Physical Changes |
| 7 | Trust |
| 9 | Reliable Communication |

## 2.4 WP4: Experiments and Testing

### Introductory remarks.

The main focus of the fourth workpackage of FRONTS is experiments and testing. The goal of these experiments is to demonstrate the practical use of theoretically sound models and algorithms from WP1, WP2, and WP3. And, more important, we expect the experiments to give valuable feedback to algorithmic methods, and even allows to discover new challenges that occur when bringing a theoretical concept into practical use.

This work is done in two ways: First, there are simulations and experiments that are carried out by one or two groups. These activities focus on certain aspects and specific problems, and use local expertise and locally existing soft- and hardware. Second, there are unifying experiments that span a larger area and are done by at least three partners. Our approach is to test algorithms and models by simulations to obtain first results. Preferably, we use the network simulator *Shawn* (`http://shawn.sourceforge.net`). Afterwards, we want to move on to testing on real sensor networks, for example the testbed in Lübeck. Some experiments have been done already in real hardware; for example, the Load Balancing and Neighborhood discovery experiments are done on Sun SPOTS. In WP2 and WP3, there are unifying use case scenarios (see D2.2, D3.2). The experiments presented here cover some of the aspects of these scenarios. To coordinate the efforts of all project partners, we set up a central repository for experiments, containing the software components as well as documentations.

So far, we have 9 standalone experiments and 7 unifying experiments spreading a large variety of important challenges in adaptive network societies —such as security of RFIDs—, formal methods such as population protocols, and interesting applications (e.g., animal tracking).

## Report on the Experiment Repository.

The Central Experiments Repository of algorithms and experiments provides a basis to coordinate the activities of the project partners. We here report the implemented software and the corresponding documentation produced by the project partners during the 2nd year of FRONTS. Technical details on the code repository and experimental environment setup and operation guidelines are available in D4.2.

The experimental work conducted in FRONTS is documented in 26 technical reports. 12 technical reports were produced during the 1st year of the project and 14 technical reports were produced during the 2nd year. Furthermore 7 well designed unifying experiments were conducted.

These studies produced 25 software components that are implemented following three different approaches: (A) in specific simulation platforms (e.g., ns-2, matlab, OMNeT++) or specific hardware platforms (e.g., SUN SPOT, TinyOS) or stand-alone software (e.g., C++, Java, Python, Ruby); (B) compatible with the centrally coordinated simulation test bed, and (C) compatible with the centrally coordinated experimentation test bed.

| **Component** | **Approach** | **WP** | **Related TR** |
|---|---|---|---|
| AB_secure_transmitting<br>*Protocol for secure routing* | A (TinyOS) | WP3 | - |
| AggregateMax<br>*Window streaming algorithms for data aggregation* | A (Python) | WP1 | - |
| | *continues below* | | |

| Component | Approach | WP | Related TR |
|---|---|---|---|
| APSR<br>*Protocol for adaptive secure routing* | B | WP3 | 2008-9 |
| BP-VER<br>*Verifiers for Population Protocols* | A (C++) | WP1 | - |
| BSM<br>*Protocols for sink mobility* | A (ns-2) | WP3 | 2009-10 |
| Cooperation<br>*Framework for evaluating trust and cooperation arising from interactions* | B | WP3 | - |
| DutyCycling<br>*Mechanism for self-synchronized duty-cycling* | A (C++) | WP1 | 2009-54 |
| FairPropagation<br>*Protocols for fair propagation* | A (C++) | WP2 | 2008-6 |
| FinN<br>*Framework for pervasive games and interactive installations* | A (SUN SPOT) | WP3 | 2009-58, 2009-31, 2008-35, 2008-28, 2008-27 |
| FunctionalMonitoring<br>*Application for Functional Monitoring* | B | WP2 | - |
| LoadBalancing<br>*Protocols for load and communication balancing* | A (SUN SPOT) | WP2 | - |
| MapReduce<br>*Framework for implementing MapReduce applications* | B | WP2 | - |
| MobiShawn<br>*Framework for drove monitoring* | B | WP3 | - |
| NewRandomForest<br>*Method for corruption of existing data - Multi Feature Learning* | A (Matlab) | WP2 | 2008-50 |
| OA<br>*Protocols for routing with obstacle avoidance* | A (Matlab) | WP3 | 2009-13 |
| PassiveTargetTracking<br>*Algorithms for target tracking* | A (Ruby) | WP1 | - |
| PopulationProtocols<br>*Simulation framework for Population Protocols* | A (Java) | WP1 | 2009-3 |
| RecommendationSimulator<br>*Fully decentralized recommendation system* | A (Matlab) | WP2 | 2010-3 |
| RecommendationSMS<br>*Lightweight SMS-based recommendation system for mobile users* | A (Java) | WP2 | 2010-6 |
| RemoveAdditionByzantineData<br>*Method for identifying and filtering Byzantine Data* | A (Matlab) | WP2 | 2008-50 |
| RobotControl<br>*Protocols for robot coordination* | A (SUN SPOT) | WP3 | - |
| SecureRFID<br>*Framework for privacy for RFID* | B | WP2 | - |
| StarFileAllocation<br>*Algorithms for power-aware online file allocation* | B | WP2 | 2009-20 |
| SSO<br>*Protocols for self-organization of the infrastructure* | B | WP2 | - |
| TDS<br>*Algorithms for duplicate insensitive counting* | A (C++) | WP1 | 2009-19 |
|  | *End of table* |  |  |

## Software Components:

- Total of 25 components currently

- 8 using Shawn
- 4 using Matlab
- 4 already running on WSN hardware
- By WP: WP1 6; WP2 11; WP3 8
- Code adheres to FRONTS coding standards (except for build.xml)

**Our unifying experiments.**

The *first* unifying experiment deals with data aggregation in a window streaming model, where there is less storage than needed to perform an exact computation. The *second* considers population protocols, a model for distributed computing using nodes whose capabilities are equivalent to finite state machines. Security of RFIDs (e.g., who is allowed to read what RFID tag?) is investigated in the *third* experiment, followed by aspects of self-organization in a network, such as leader election, which is the *fourth* experiment. The *fifth* experiment deals with security, again: secure routing protocols in wireless sensor networks. Tracking agents in a network using scent-like traces is the topic of the *sixth* experiment. The *seventh* experiment deals with distributing files in a WSN; for example, to flash every node.

The relation of our 7 unifying experiments to partners involved and to workpackages is shown in Figure 2. One can see the integration achieved!

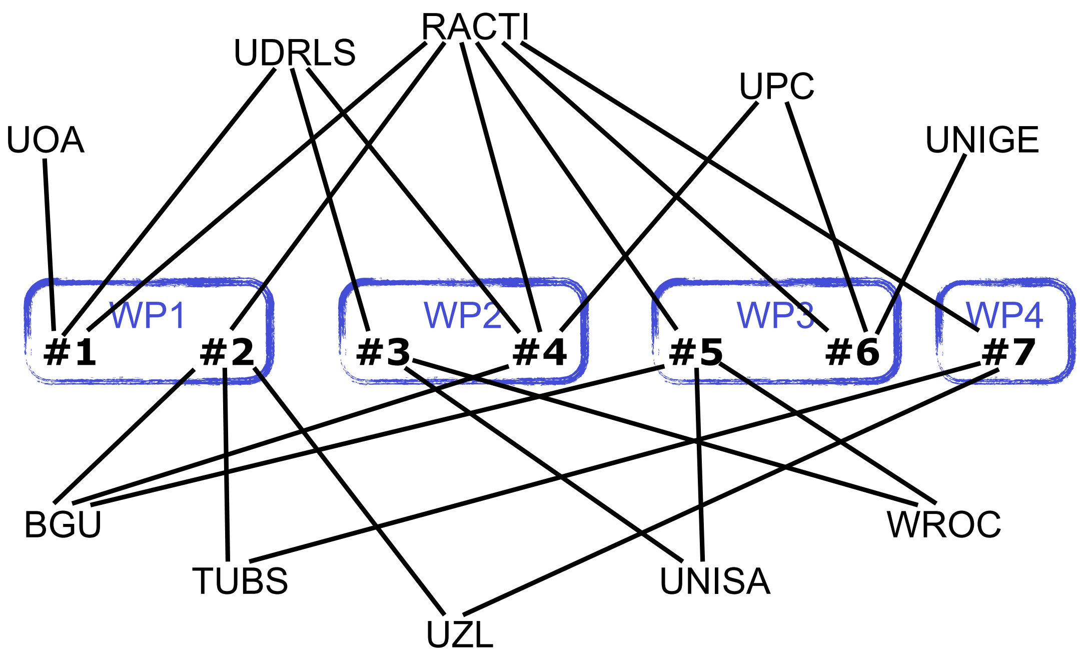

Figure 2: Our 7 experiments and unification

## Our "standalone" experiments.

These experiments were done by one (or two) partner(s) in order to test schemes that were left out in our 7 unifying experiments. We table them in Table 8.

Table 8: Standalone experiments

| Experiment | Partners | WP |
|---|---|---|
| Duplicate Insensitive Counting | RACTI, UPC | WP1 |
| Decentralized Recommendation Systems | UDRLS | WP2 |
| File Allocation | UPB | WP2 |
| Duty Cycling | UPC | WP3 |
| Load & Communication Balancing | UPC | WP3 |
| Neighborhood Discovery & DTN | RACTI | WP3 |
| Obstacle Avoidance | RACTI | WP3 |
| Fast Data Collection | RACTI | WP3 |
| Trust and Cooperation | UNISA | WP3 |

## Some remarks.

In the 2nd year, we got a rich set of experiments. Most of them validate our schemes. A rich first set of conclusions per experiment has been drawn and we have to systematize these conclusions in the 3rd year. In addition, we have started to form some practical notions of "what to measure". A partial list is given in Table 9. In the 3rd year we will prioritize and complete the set of these measures.

Table 9: Measures vs functionality

| Functionality | Measures |
|---|---|
| Protocols verification | ▷ Population size<br>▷ Number of states |
| RFID security | ▷ Number of tags<br>▷ Range of tracers |
| Self-organization | ▷ Response time |
| Secure routing | ▷ Redundancy of routes versus faults rate |
| Tracking | ▷ Speed/form of dynamic changes versus efficiency of tracking |
| Data aggregation | ▷ Memory needed<br>▷ Competitive ratios |
| Software dissemination | ▷ Concurrency<br>▷ Scalability<br>▷ Feedback |

## 3 Some Conclusions

1. Our implemented Software modules have shown the “common denominator” techniques needed in many schemes.
2. We have increased cooperation and the common view to the methods and goals of FRONTS via our joint experimental work.
3. Some of our experiments are in a very preliminary stage. We have to continue and enrich the possible input scenaria.
4. Our models seem to mature. Perhaps we can unify (or at least compare) EPP and SFA.
5. We have a preliminary idea of what are the important measures per scheme for measuring performance. We should complete and clean our measures.

## References

The union of references are included in the 4 WP reports (deliverables D1.2, D2.2, D3.2, D4.4). We do not repeat them here.