*Ruben Smoor, Inti Ridder, Frank Montenij, Martijn Lammers*

*Automated Systems, Hogeschool Rotterdam*

*Technical Documentation*

Swarm Robotic Models

# SSS: System Subsystem Specification

## Scope.

*The goal of the project is to mimic a natural occurring flock of coordinated insects or animals. Think of a group of ants who all coordinate together to find, notify and carry food for the tribe.*

*Our Robotic Swarm wants to achieve the following goal: the RS has to cooperate to find a target by moving in an undiscovered flat environment while maintaining to dodge fellow robots and obstacles. Once the target has been found by one of the robots, the other robots are going to be notified and the RB will surround the target as a group.*

De swarm zal een object moeten zoeken in een omgeving met ‘obstakels’. De positiebepaling zal indien mogelijk door middel van ultrasone sensoren en signalen werken, soortgelijk aan het GPS-systeem. Met als optioneel onderdeel het omsingelen van het object waneer het gevonden is.

Must haves:

* De Swarm robots moeten kunnen communiceren met een server.
* Er is een simulatie die hetzelfde gedrag als de fysieke modellen nabootst.
* De simulatie moet samenwerken met het fysieke model.
* De robots moeten een doel zoeken.
* De robots moeten informatie opsturen naar de server als een doel gevonden is.
* De robots laten zien wat ze doen door middel van LEDs
* De robots moeten individueel obstakels detecteren en vermijden.
* De robots moeten hun positie kunnen bepalen.

Should haves:

* De positie wordt bepaald in 2D door middel van ultrasone signalen en sensoren soortgelijk aan het GPS-systeem.
* De nodes moeten een doel omsingelen als het gevonden is.

## System overview.

*This paragraph shall describe the general structure of the system, and for each part a brief description of the scope of that part.*

*The RS has the following components:*

* *Server: The server is written in Javascript using NodeJS. The Server communicates with the other components using a MQTT broker. The broker is the supervisor that orchestrates the instructions of the physical and simulated swarm models using various algorithms such as shortest path and grid search.*
* *Physical model: The physical model is built using Lego bricks as a chassis. The chassis holds two EV3 servomotors, two microcontrollers (Arduino Nano, Microstorm board v2), a battery and sensors (LDR and Infrared). The microcontrollers both run C/C++ Arduino programs that can influence the actuators, read sensors and communicate data from and to the observer-client and server.*
* *Simulation model: The simulation is used as a testing environment and backup proof of concept of the physical model. The simulation is made in Webots, the world consists of a 10x10 node grid. A single node is square of 20cm^2. The simulated obstacles are represented by black nodes and the target consists of a blue 10cm^3 cube with a light source. The simulated robot is a simplified shaped version of the physical model with two wheels, a stabilization leg and sensors (LDR and Infrared). The simulated robot does not mimic the battery, microcontrollers or connectors as these are irrelevant additions to simulate because they do not influence the behaviour of the robot movement or it’s logic.*
* *Simulation controller: As the name suggest, the controller is used to control the Webots robots. The controller is written in Python for easy prototyping and dependency management. Just as the physical microcontrollers, the Python script is used to move the robot, read sensor data and communicate with the server using MQTT.*
* *Observer-client: The observer is used to supplement the server with information of the physical robots that they can not offer themselves. The observer is solely responsible for tracking the physical robot and sending the position and heading to the server. The observer is written in Python and uses OpenCV to track the robots.*
* *Playground:*
* *Obstacle*

# SSDD: System Subsystem Design Description

## System-wide design decisions.

*This section shall be divided into paragraphs as needed to present system-wide design decisions, that is, decisions about the system's behavioral design (how it will behave, from a user's point of view, in meeting its requirements, ignoring internal implementation) and other decisions affecting the selection and design of system components. If all such decisions are explicit in the requirements or are deferred to the design of the system components, this section shall so state. Design decisions that respond to requirements designated critical, such as those for safety, security, or privacy, shall be placed in separate subparagraphs. If a design decision depends upon system states or modes, this dependency shall be indicated. Design conventions needed to understand the design shall be presented or referenced.*

*All decisions shall state all considered alternatives and be supported by references to literature.*

*Examples of system-wide design decisions are the following:*

* *Design decisions regarding inputs the system will accept and outputs it will produce, including interfaces with other systems, configuration items, and. If part or all of this information is given in Interface Design Descriptions (IDDs), they may be referenced.*
* *Design decisions on system behavior in response to each input or condition, including actions the system will perform, response times and other performance characteristics, description of physical systems modeled, selected equations/algorithms/ rules, and handling of unallowed inputs or conditions.*
* *Design decisions on how system databases/data files will appear to the user (4.3.x of this DID identifies topics to be considered in this description). If part or all of this information is given in Database Design Descriptions (DBDDs), they may be referenced.*
* *Selected approach to meeting safety, security, and privacy requirements.*
* *Design and construction choices for hardware or hardware-software systems, such as physical size, color, shape, weight, materials, and markings.*
* *Other system-wide design decisions made in response to requirements, such as selected approach to providing required flexibility, availability, and maintainability.*

## System architectural design.

*This section shall be divided into the following paragraphs to describe the system architectural design. If part or all of the design depends upon system states or modes, this dependency shall be indicated. If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs. Design conventions needed to understand the design shall be presented or referenced.   
  
Note: For brevity, this section is written in terms of organizing a system directly into Hardware Configuration Items (HWCIs), Computer Software Configuration Items (CSCIs), and manual operations, but should be interpreted to cover organizing a system into subsystems, organizing a subsystem into HWCIs, CSCIs, and manual operations, or other variations as appropriate.*

## System components.

*This paragraph shall:*

1. *Identify the components of the system (HW and SW). Each component shall be assigned a name.*
2. *Show the static relationship(s) of the components in a diagram.*
3. *State the purpose of each component and identify the system-wide design decisions allocated to it.*

# SDD: Software Design Description

*Fill in only if further design of subsystems is required (think of class diagram, sequence diagram etc of a subsystem)*

# SRS: System Requirements Specification

## Requirements.

*This section shall be divided into the following paragraphs to specify the system requirements, that is, those characteristics of the system that are conditions for its acceptance. Each requirement shall be assigned a project-unique identifier to support testing and traceability and shall be stated in such a way that an objective test can be defined for it. Each requirement shall be annotated with associated qualification method(s) (see section 4) and, for subsystems, traceability to system requirements (see section 5.a), if not provided in those sections. The degree of detail to be provided shall be guided by the following rule: Include those characteristics of the system that are conditions for system acceptance; defer to design descriptions those characteristics that the acquirer is willing to leave up to the developer. If there are no requirements in a given paragraph, the paragraph shall so state. If a given requirement fits into more than one paragraph, it may be stated once and referenced from the other paragraphs.*

## Required states and modes.

*If the system is required to operate in more than one state or mode having requirements distinct from other states or modes, this paragraph shall identify and define each state and mode. Examples of states and modes include: idle, ready, active, post-use analysis, training, degraded, emergency, backup, wartime, peacetime. The distinction between states and modes is arbitrary. A system may be described in terms of states only, modes only, states within modes, modes within states, or any other scheme that is useful. If no states or modes are required, this paragraph shall so state, without the need to create artificial distinctions. If states and/or modes are required, each requirement or group of requirements in this specification shall be correlated to the states and modes. The correlation may be indicated by a table or other method in this paragraph, in an appendix referenced from this paragraph, or by annotation of the requirements in the paragraphs where they appear.*

## System capability requirements.

*This paragraph shall be divided into subparagraphs to itemize the requirements associated with each capability of the system. A "capability" is defined as a group of related requirements. The word "capability" may be replaced with "function," "subject," "object," or other term useful for presenting the requirements.*

### (System capability).

*This paragraph shall identify a required system capability and shall itemize the requirements associated with the capability. If the capability can be more clearly specified by dividing it into constituent capabilities, the constituent capabilities shall be specified in subparagraphs. The requirements shall specify required behavior of the system and shall include applicable parameters, such as response times, throughput times, other timing constraints, sequencing, accuracy, capacities (how much/how many), priorities, continuous operation requirements, and allowable deviations based on operating conditions. The requirements shall include, as applicable, required behavior under unexpected, unallowed, or "out of bounds" conditions, requirements for error handling, and any provisions to be incorporated into the system to provide continuity of operations in the event of emergencies*

## System external interface requirements.

*This paragraph shall be divided into subparagraphs to specify the requirements, if any, for the system's external interfaces. This paragraph may reference one or more Interface Requirements Specifications (IRSs) or other documents containing these requirements.*

## System internal interface requirements.

*This paragraph shall specify the requirements, if any, imposed on interfaces internal to the system. If all internal interfaces are left to the design or to requirement specifications for system components, this fact shall be so stated. If such requirements are to be imposed, paragraph 3.3 of this DID provides a list of topics to be considered.*

## System internal data requirements.

This paragraph shall specify the requirements, if any, imposed on data internal to the system. Included shall be requirements, if any, on databases and data files to be included in the system. If all decisions about internal data are left to the design or to requirements specifications for system components, this fact shall be so stated. If such requirements are to be imposed, paragraphs 3.3.x.c and 3.3.x.d of this DID provide a list of topics to be considered.

## Adaptation requirements.

This paragraph shall specify the requirements, if any, concerning installation-dependent data that the system is required to provide (such as site-dependent latitude and longitude or site-dependent state tax codes) and operational parameters that the system is required to use that may vary according to operational needs (such as parameters indicating operation-dependent targeting constants or data recording).

## Safety requirements.

This paragraph shall specify the system requirements, if any, concerned with preventing or minimizing unintended hazards to personnel, property, and the physical environment. Examples include restricting the use of dangerous materials; classifying explosives for purposes of shipping, handling, and storing; abort/escape provisions from enclosures; gas detection and warning devices; grounding of electrical systems; decontamination; and explosion proofing. This paragraph shall include the system requirements, if any, for nuclear components, including, as applicable, requirements for component design, prevention of inadvertent detonation, and compliance with nuclear safety rules.

## Security and privacy requirements.

This paragraph shall specify the system requirements, if any, concerned with maintaining security and privacy. The requirements shall include, as applicable, the security/privacy environment in which the system must operate, the type and degree of security or privacy to be provided, the security/privacy risks the system must withstand, required safeguards to reduce those risks, the security/privacy policy that must be met, the security/privacy accountability the system must provide, and the criteria that must be met for security/privacy certification/accreditation.

## System environment requirements.

This paragraph shall specify the requirements, if any, regarding the environment in which the system must operate. Examples for a software system are the computer hardware and operating system on which the software must run. (Additional requirements concerning computer resources are given in the next paragraph). Examples for a hardware-software system include the environmental conditions that the system must withstand during transportation, storage, and operation, such as conditions in the natural environment (wind, rain, temperature, geographic location), the induced environment (motion, shock, noise, electromagnetic radiation), and environments due to enemy action (explosions, radiation).

## System quality factors.

This paragraph shall specify the requirements, if any, pertaining to system quality factors. Examples include quantitative requirements concerning system functionality (the ability to perform all required functions), reliability (the ability to perform with correct, consistent results -- such as mean time between failure for equipment), maintainability (the ability to be easily serviced, repaired, or corrected), availability (the ability to be accessed and operated when needed), flexibility (the ability to be easily adapted to changing requirements), portability of software (the ability to be easily modified for a new environment), reusability (the ability to be used in multiple applications), testability (the ability to be easily and thoroughly tested), usability (the ability to be easily learned and used), and other attributes.

## Design and construction constraints.

*This paragraph shall specify the requirements, if any, that constrain the design and construction of the system. For hardware-software systems, this paragraph shall include the physical requirements imposed on the system. These requirements may be specified by reference to appropriate commercial or military standards and specifications. Examples include requirements concerning:*

1. *Use of a particular system architecture or requirements on the architecture, such as required subsystems; use of standard, military, or existing components; or use of Government/acquirer-furnished property (equipment, information, or software)*
2. *Use of particular design or construction standards; use of particular data standards; use of a particular programming language; workmanship requirements and production techniques*
3. *Physical characteristics of the system (such as weight limits, dimensional limits, color, protective coatings); interchangeability of parts; ability to be transported from one location to another; ability to be carried or set up by one, or a given number of, persons*
4. *Materials that can and cannot be used; requirements on the handling of toxic materials; limits on the electromagnetic radiation that the system is permitted to generate*
5. *Use of nameplates, part marking, serial and lot number marking, and other identifying markings*
6. *Flexibility and expandability that must be provided to support anticipated areas of growth or changes in technology, threat, or mission*

## Other requirements.

*This paragraph shall specify additional system requirements, if any, not covered in the previous paragraphs. Examples include requirements for system documentation, such as specifications, drawings, technical manuals, test plans and procedures, and installation instruction data, if not covered in other contractual documents.*

# IDD: Interface Design Description

## Interface design.

*This paragraph shall be divided into the following subparagraphs to describe the interface characteristics of the system components. It shall include both interfaces among the components and their interfaces with external entities such as other systems, configuration items, and users. One or more interface diagrams shall be provided, as appropriate, to depict the interfaces.*

## Interface identification and diagrams.

For each interface identified in 5.1 , this paragraph shall state the project-unique identifier assigned to the interface and shall identify the interfacing entities (systems, configuration items, users, etc.) by name, number, version, and documentation references, as applicable. The identification shall state which entities have fixed interface characteristics (and therefore impose interface requirements on interfacing entities) and which are being developed or modified (thus having interface requirements imposed on them). One or more interface diagrams shall be provided, as appropriate, to depict the interfaces.

### (Project unique identifier of interface).

*This paragraph (beginning with 3.2) shall identify an interface by project unique identifier, shall briefly identify the interfacing entities, and shall be divided into subparagraphs as needed to describe the interface characteristics of one or both of the interfacing entities. If a given interfacing entity is not covered by this IDD (for example, an external system) but its interface characteristics need to be mentioned to describe interfacing entities that are, these characteristics shall be stated as assumptions or as "When [the entity not covered] does this, [the entity that is covered] will ...." This paragraph may reference other documents (such as data dictionaries, standards for protocols, and standards for user interfaces) in place of stating the information here. The design description shall include the following, as applicable, presented in any order suited to the information to be provided, and shall note any differences in these characteristics from the point of view of the interfacing entities (such as different expectations about the size, frequency, or other characteristics of data elements):*

1. *Priority assigned to the interface by the interfacing entity(ies)*
2. *Type of interface (such as real-time data transfer, storage-and-retrieval of data, etc.) to be implemented*
3. *Characteristics of individual data elements that the interfacing entity(ies) will provide, store, send, access, receive, etc., such as:* 
   1. *Names/identifiers* 
      1. *Project-unique identifier*
      2. *Non-technical (natural-language) name*
      3. *DoD standard data element name*
      4. *Technical name (e.g., variable or field name in code or database)*
      5. *Abbreviation or synonymous names*
   2. *Data type (alphanumeric, integer, etc.)*
   3. *Size and format (such as length and punctuation of a character string)*
   4. *Units of measurement (such as meters, dollars, nanoseconds)*
   5. *Range or enumeration of possible values (such as 0-99)*
   6. *Accuracy (how correct) and precision (number of significant digits)*
   7. *Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the data element may be updated and whether business rules apply*
   8. *Security and privacy constraints*
   9. *Sources (setting/sending entities) and recipients (using/receiving entities)*
4. *Characteristics of data element assemblies (records, messages, files, arrays, displays, reports, etc.) that the interfacing entity(ies) will provide, store, send, access, receive, etc., such as:* 
   1. *Names/identifiers* 
      1. *Project-unique identifier*
      2. *Non-technical (natural language) name*
      3. *Technical name (e.g., record or data structure name in code or database)*
      4. *Abbreviations or synonymous names*
   2. *Data elements in the assembly and their structure (number, order, grouping)*
   3. *Medium (such as disk) and structure of data elements/assemblies on the medium*
   4. *Visual and auditory characteristics of displays and other outputs (such as colors, layouts, fonts, icons and other display elements, beeps, lights)*
   5. *Relationships among assemblies, such as sorting/access characteristics*
   6. *Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the assembly may be updated and whether business rules apply*
   7. *Security and privacy constraints*
   8. *Sources (setting/sending entities) and recipients (using/receiving entities)*
5. *Characteristics of communication methods that the interfacing entity(ies) will use for the interface, such as:* 
   1. *Project-unique identifier(s)*
   2. *Communication links/bands/frequencies/media and their characteristics*
   3. *Message formatting*
   4. *Flow control (such as sequence numbering and buffer allocation)*
   5. *Data transfer rate, whether periodic/aperiodic, and interval between transfers*
   6. *Routing, addressing, and naming conventions*
   7. *Transmission services, including priority and grade*
   8. *Safety/security/privacy considerations, such as encryption, user authentication, compartmentalization, and auditing*
6. *Characteristics of protocols the interfacing entity(ies) will use for the interface, such as:* 
   1. *Project-unique identifier(s)*
   2. *Priority/layer of the protocol*
   3. *Packeting, including fragmentation and reassembly, routing, and addressing*
   4. *Legality checks, error control, and recovery procedures*
   5. *Synchronization, including connection establishment, maintenance, termination*
   6. *Status, identification, and any other reporting features*
7. *Other characteristics, such as physical compatibility of the interfacing entity(ies) (dimensions, tolerances, loads, voltages, plug compatibility, etc.)*

# STP: System Test Plan

## Scope.

*This section shall be divided into the following paragraphs.*

## Software test environment.

*This section shall be divided into the following paragraphs to describe the software test environment at each intended test site. Reference may be made to the Software Development Plan (SDP) for resources that are described there.*

## Software items.

*This paragraph shall identify by name, number, and version, as applicable, the software items (e.g., operating systems, compilers, communications software, related applications software, databases, input files, code auditors, dynamic path analyzers, test drivers, preprocessors, test data generators, test control software, other special test software, post processors) necessary to perform the planned testing activities at the test site(s). This paragraph shall describe the purpose of each item, describe its media (tape, disk, etc.), identify those that are expected to be supplied by the site, and identify any classified processing or other security or privacy issues associated with the software items.*

## Hardware and firmware items.

*This paragraph shall identify by name, number, and version, as applicable, the computer hardware, interfacing equipment, communications equipment, test data reduction equipment, apparatus such as extra peripherals (tape drives, printers, plotters), test message generators, test timing devices, test event records, etc., and firmware items that will be used in the software test environment at the test site(s). This paragraph shall describe the purpose of each item, state the period of usage and the number of each item needed, identify those that are expected to be supplied by the site, and identify any classified processing or other security or privacy issues associated with the items.*

## Other materials.

*This paragraph shall identify and describe any other materials needed for the testing at the test site(s). These materials may include manuals, software listings, media containing the software to be tested, media containing data to be used in the tests, sample listings of outputs, and other forms or instructions. This paragraph shall identify those items that are to be delivered to the site and those that are expected to be supplied by the site. The description shall include the type, layout, and quantity of the materials, as applicable. This paragraph shall identify any classified processing or other security or privacy issues associated with the items.*

## Test identification.

*This section shall be divided into the following paragraphs to identify and describe each test to which this STP applies.*

### General information.

*This paragraph shall be divided into subparagraphs to present general information applicable to the overall testing to be performed.*

### General test conditions.

*This paragraph shall describe conditions that apply to all of the tests or to a group of tests. For example: "Each test shall include nominal, maximum, and minimum values;" "each test of type x shall use live data." Included shall be a statement of the extent of testing to be performed and rationale for the extent selected. The extent of testing shall be expressed as a percentage of some well defined total quantity, such as the number of samples of discrete operating conditions or values, or other sampling approach. Also included shall be the approach to be followed for retesting/regression testing.*

### Test progression.

*In cases of progressive or cumulative tests, this paragraph shall explain the planned sequence or progression of tests.*

### Data recording, reduction, and analysis.

*This paragraph shall identify and describe the data recording, reduction, and analysis procedures to be used during and after the tests identified in this STP. These procedures shall include, as applicable, manual, automatic, and semi-automatic techniques for recording test results, manipulating the raw results into a form suitable for evaluation, and retaining the results of data reduction and analysis.*

## Planned tests.

*This paragraph shall be divided into the following subparagraphs to describe the total scope of the planned testing.*

### (Project-unique identifier of a test).

*This paragraph shall identify a test by project unique identifier and shall provide the information specified below for the test.*

1. *Test objective*
2. *Test level*
3. *Test type or class*
4. *Qualification method(s) as specified in the requirements specification*
5. *Identifier of the CSCI requirements and, if applicable, software system requirements addressed by this test. (Alternatively, this information may be provided in Section 6.)*
6. *Special requirements (for example, 48 hours of continuous facility time, weapon simulation, extent of test, use of a special input or database)*
7. *Type of data to be recorded*
8. *Type of data recording/reduction/analysis to be employed*
9. *Assumptions and constraints, such as anticipated limitations on the test due to system or test conditions--timing, interfaces, equipment, personnel, database, etc.*
10. *Safety, security, and privacy considerations associated with the test*

# STR: System Test Report

## Overview of test results.

*For each paragraph 6.7 test results will be reported here.*

### (Project-unique identifier of a test).

*Test results of this test*

## Test log.

*This section shall present, possibly in a figure or appendix, a chronological record of the test events covered by this report. This test log shall include:*

1. *The date(s), time(s), and location(s) of the tests performed*
2. *The hardware and software configurations used for each test including, as applicable, part/model/serial number, manufacturer, revision level, and calibration date of all hardware, and version number and name for the software components used*
3. *The date and time of each test related activity, the identity of the individual(s) who performed the activity, and the identities of witnesses, as applicable*

# Notes.

*This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.*

# Referenced documents.

*This section shall list the number, title, revision, and date of all documents referenced in this specification. This section shall also identify the source for all documents not available through normal Government stocking activities.*

# A. Appendixes.

*Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).*