

Systems grammar in OpenSpecs

Status of document: **Frozen for implementation**

Authors list

Name	Organization
Ingmar Ogren	ROMET AB
Eric Verhulst	Open License Society
Vitaliy Mezhuyev	Berdiansk State Pedagogical University, Department of Computer Systems and Networks
Gjalt de Jong	Melexis
Nikolay Nikitchenko	Kiev National University, Department of theory and technology of programming

Document history

Date	Version	Author	Change
26-Apr-04	1.0.0	Ingmar Ogren	Definition of systems grammar of Tofs
22-Mar-05	1.0.5	Eric Verhulst	Definition of systems grammar of OpenSpecs
27-Mar-05	1.0.6	Vitaliy Mezhuyev	Definition of base principles, editing
28-Mar-05	1.0.7	Vitaliy Mezhuyev	Editing
04-Apr-05	1.0.8	Vitaliy Mezhuyev	Editing
15-Apr-05	1.0.8	Eric Verhulst Gjalt de Jong	Refining
17-Apr-05	1.0.9	Vitaliy Mezhuyev	Editing
18-Sep-05	1.1.0	Vitaliy Mezhuyev	Editing
08-Oct-05	1.1.1	Vitaliy Mezhuyev, Nikolay Nikitchenko	Editing, modification of principles of system grammar formulation
09-Oct-05	1.1.1	Gjalt de Jong	Editing
10-Oct-05	1.1.1	Eric Verhulst	Editing
13-Oct-05	1.1.2	Vitaliy Mezhuyev	Editing
27-Oct-05	1.1.3	Eric Verhulst, Gjalt de Jong, Vitaliy Mezhuyev	Clarifications and simplifications
27-Oct-05	1.1.3	Gjalt de Jong	Editing
30-Oct-05	1.1.4	Vitaliy Mezhuyev	Generalisations
31-Oct-05	1.1.4b	Eric Verhulst	Editing
01-Nov-05	1.1.4c	Gjalt de Jong	Editing
06-Nov-05	1.1.5	Vitaliy Mezhuyev	Version for implementation
26-Mar-06	1.1.6	Vitaliy Mezhuyev	Last edition



Contents

Authors list	1
Document history	1
Contents.....	2
General principles of the Systems grammar definition	3
General definitions	4
SYSTEMS GRAMMAR	6
Project view	6
This grammar rules relates to the scope definition phase of SE process and defines the context of the system (under development).Architectural view	6
Architectural view	7
Requirements view	8
Transition from requirements to architectural level	10
Planning view	11
The intentional (target) approach for entity, system and project definition	12
Structure of requirements at the Project level	13
Definitions of the Systems grammar concepts	14
The Definitions of Systems Grammar Terms accordingly to the Britannica dictionary (www.britannica.com).....	18

General principles of the Systems grammar definition

The Systems grammar in OpenSpecs is based on a unified and coherent methodology:

1. Systems engineering domain.
2. Interacting entities paradigm.
3. Declarative (intentional) and imperative (extensional) paradigms of models development.

General principles of the Systems grammar definition:

1. The Systems grammar defines a model at **three levels**: requirements (declarative or intentional), architectural (imperative or extensional) and also planning views.
2. The declarative level means the **description** of the system properties, its structure and functions in a natural language (metamodel development).
3. The imperative level means the **definition** of the system properties, its structure and functions in a formalized language (model development).

OpenSpecs is a tool for transformation of the system description into the system definition.

For the system description and definition cognitive (gnosiological) and ontological concepts are used.

1. The ontology domain is reflected by vocabulary of notions such as system, entity, interaction, etc. and also relations terms such as be, have, consist of, etc.
2. At the cognitive level a system is reflected by notions such as requirement, specification, test case etc. and also relations terms such as defined by, consist of, achieve, descendant of etc.
3. The cognitive (conceptual) level is specified first, because requirements are mapped into entities and their interactions.
4. The transitions between the cognitive and the ontological levels are achieved by decomposition, typification and structuring methods.
5. The results of the requirements decomposition serve as measures for defining specifications, test cases, failure cases.

General definitions

Each concept of the Systems grammar is an attribute of a project:

Concept IS_AN

Attribute

Each concept of the Systems grammar is defined by own attributes (e.g. description, status or issue of project development):

Concept (1-N) IS_DEFINED_BY

Attribute (1-N)

Each attribute has an obligatory identifier, value and is defined by own type.

Attribute IS_DEFINED_BY

Identifier (1) AND

// Title or Name

Type (1) AND

Value (1)

ConceptsTypes ARE

// Hierarchy of content types

Project OR

ReqSpecs OR

Requirement OR

Specification OR

NormalCase OR

TestCase OR

FaultCase OR

Issue OR

Architecture OR

Entity OR

Attribute OR

Function OR

Interface OR

Interaction OR

WorkPlan OR

WorkPackage OR

ChangeRequest OR

Task

AttributeType IS

// Possible attributes of System Grammar
concepts

Boolean OR

Character OR

Integer OR

Float OR

DateTime OR

String OR

Description OR

// Text. Used in more general cases

Structured OR

// I.e. Array, Record, Set, etc.

Status OR

// Reflects progress in time of concepts development

Issue OR // Obligatory field of any concept



Specific // I.e. user defined

Each concept has obligatory attributes – Description and Status.

Description IS_DEFINED_BY

Source AND // I.e. author or citing

CreationDateTime AND

StructuredText AND

ModificationsHistory // I.e. reviewers list, dates, changes and its motivations

Description CAN_BE

Text OR // For Requirements and Specifications

Purpose OR // For entities

Mission OR // For systems

Objective OR // For projects

Resolution OR // For issues

Motivation OR // For issues

Deliverables OR // For Tasks

Resources OR // For Tasks

Responsible OR // For Tasks

Dependencies // For WP

Status reflects progress in time of concepts development

Status IS

In_Work OR // On progress of elaboration

Frozen_for_Review OR // To be discussed before approving

Frozen_and_Approved // Accepted as system attribute

SYSTEMS GRAMMAR

Project view

The Systems grammar provides a framework to describe the project in three orthogonal directions (views):

Project HAS_VIEWS

Requirements_View (1) AND
Architectural_View (1) AND
Planning_View (1)

The Requirements views includes: Requirements, Specifications, Test cases, and Failure cases as views on the system under development, its operator or the environment that are also considered as systems.

The Architectural view defines: Entities, their Attributes, Functions, Interfaces and Interactions. The planning view defines sets of Work Packages, Changes Requests and Tasks for the system development.

Project DEFINES

System (1) AND
Environment (1-N) AND
Operator (1-N)

The term „System“ denotes „System-under-development“ unless stated otherwise.

Environment (1-N) and Operator (1-N) means that one system (1) can be used in many (from 1 to N) environments by different (from 1 to N) operators.

This grammar rules relates to the scope definition phase of SE process and defines the context of the system (under development).

Architectural view

The main structural part of a system is an entity.

Entity IS_DEFINED_BY

Purpose (1) AND	
DevelopmentStatus (1) AND	
Issues (1) AND	
Functions (1-N) AND	
Interfaces (1-N) AND	// Can be of internal (for entity) or external (for system) type
Entities (0-N) AND	
Interactions (0-N) AND	// E.g. between enclosing entities or system and environment (and operator)
Attributes (1-N)	// Additional architectural attributes

System IS_AN Entity

The difference between a System and an Entity is mostly related to the level at which it is considered. A System is an emerging property of Entities that interact. The “Purpose” of a System is also called “Mission”. The minimum system has two entities¹ and one interaction between them. So a system is defined by own structure i.e. relations between its constituting elements. We call these the *interactions* between the entities. E.g. the system 'door opener' needs operator (entity1), 'Door' (entity2) and the definition of the interaction 'Open'.

Function IS_DEFINED_BY

Description (1) AND	
Operations (1-N) AND	
SynchronisationPoints (2-N)	// At least START and END

Interaction IS_DEFINED_BY

Structured Set of Transfers (1-N)

TransferType IS

Data OR
Energy OR
Matter

EntityDevelopmentStatus IS

PurposelIdentified OR	// Definition of identifier and purpose
AttributesIdentified OR	// Attributisation (i.e. pairs attribute–value definition)
FunctionsIdentified OR	// Behaviour definition (e.g. pairs event–method definition)
InterfacesIdentified OR	// Definition of points of interactions between an Entity and another Entity
ReadyForDesign OR	// Definition of all above levels
SpecificationsApproved OR	// Sign-off point
Implemented OR	// Entity developed as a single one
Tested OR	// Passed all acceptance tests
IntegratedIntoSystem	// Put in entities hierarchy

¹ A system, an environment and an operator we also consider as entities.

Requirements view

Requirements PRODUCE

Entities (1-N) AND
Attributes (1-N) AND
Functions (1-N) AND
Interfaces (1-N) AND
Interactions (1-N) AND
Requirements (0-N) AND
Specifications (1-N) AND
NormalCases (1-N) AND
TestCases (1-N) AND
FailureCases (1-N) AND
Issues (1-N) AND
WorkPlans (1-N) AND
R&SDocument (1)

Following order is essential.

First we define measures of a system by means of requirements decomposition into entities, their attributes, functions, interfaces and interactions.

The results of the decomposition e.g. entities and their interactions serve for specifications, test cases, work plans formulating and refining.

The end result is the generation of a Requirement and Specification document.

RSAttributes ARE

Text AND
Category AND
Importance AND
Issues AND
Status

Requirement's categories are static and dynamic requests placed on the system under development.

RSCategory IS

Control OR
Convenience OR
Economy OR
Extensibility OR
Functional OR
Maintainability OR
Performance OR
Portability OR
Quality OR
Recoverability OR
Reliability OR
Robustness OR
Safety OR // Factors that will protect project from accidental or malicious access, misuse, or modification

Scalability OR
Security OR
Structural OR // Defines structure of the system
Usability OR // Explanation what the products will and will not do
Specific

The Importance attribute is needed for a precise definition and validation of requirements categories like quality, safety, reliability etc. Importance allows formal analysis and automates decision making in case of contradictory requirements.

RSImportance IS

Mandatory OR
Recommended OR
Optional OR

Specification JUSTIFIES

Requirement (1-N)

Specification CONSISTS_OF

NormalCase (1-N) AND
TestCase (1-N) AND
FaultCase (1-N) AND
Issue (1-N)

Issue IS_DEFINED_BY

Description (1) AND
Priority (1) AND
Solution (1) AND
Motivation (1) AND
Status (1)

IssueStatus IS

Open OR
Resolved

NormalCase is a description of required system behaviour.
TestCase is needed for validation of system and issue resolution.

TestCase VERIFIES

Specification

TestCase VALIDATES

Requirement

TestCase IS_DEFINED_BY

Status (1) AND
Description (1) AND
Results (1) AND
Issues (1-N)

FaultCase IS_DEFINED_BY

Status (1) AND
Description (1) AND
FaultReasons (1-N) AND
Issues (1-N)

Transition from requirements to architectural level

Requirements DEFINE System

Requirements are suggested as the initial point of the system definition.

DEFINE means that requirements are to be decomposed into entities-interactions, which in turn have measures for definition of system properties, behaviour and structure.

Requirements result in specifications. E.g. requirement "car should be fast" results in specification "Top-speed is 160 km/hr and 0-100 km/hr acceleration in less than 10 sec".

I.e. any entity has attribute which has value of appropriate type. Thus we have entity decomposition ('car'), which proceeds in an attribute-value decomposition. In turn that's needed for specifications, test cases formulating.

Requirement CONCERNS

Purpose (1) AND	// Of entity (specific requirement) or mission of System (general requirement)
Attributes (1-N) AND	// Specific requirement
Functions (1-N) AND	// Specific requirement
Interfaces (1-N) AND	// Specific requirement
Entities (1-N) AND	// Specific requirement
Interactions (1-N) AND	// Specific requirement
System (1)	// General requirement

Any requirement, specification etc. have own object (i.e. entity, interaction etc.) as a media. During requirements definition we collect names of architectural objects: entities, interactions, etc. which are media of requirements. This is needed for feedback organization i.e. for transition from requirements to architectural levels.

Planning view

WorkPlan IS_DEFINED_BY

Milestones (2-N) AND
Deadlines (1-N) AND
Deliverables (1-N) AND
Resources (1-N) AND
Responsible (1) AND
WorkPackages (1-N) AND
ChangeRequests (1-N)

ChangeRequest IS_DEFINED_BY

Description (1) AND
Tasks (1-N)

WorkPlan PRODUCES

WorkPackages (1-N)

WorkPackage IS_DEFINED_BY

Description (1) AND
StartDate (1) AND
EndDate (1) AND
Tasks (1-N) AND
Dependencies (1-N) AND
Responsible (1)

WorkPackage PRODUCES

Tasks (1-N)

Task IS_DEFINED_BY

Description (1) AND
Priority (1) AND
Deadline (1) AND
Resources (1-N) AND
Responsible (1) AND
Deliverables (1)

WorkPackage CONCERNS

System (1)

Task CONCERNS

Entity (1-N) OR
Interaction (1-N) OR
Function (1-N) OR
Interface (1-N)

The intentional (target) approach for entity, system and project definition

Entity ACHIEVES Purpose (1)

System ACHIEVES Mission (1)

Project ACHIEVES Objective (1)

We can say about system if we have new quality of interacting entities, which each entity does not have separately. System is the highest level of interacting entities.

Integrated purposes of all entities achieve the mission of the system.

E.g. a plane is a system of interacting entities (i.e. body, wings, chassis etc.) which are separately aspiring to fall, but can fly as a whole.

Mission is the highest level of system requirements.

As entities and interactions form a system, all requirements (as implicit entities and interaction purposes) in aggregate achieve objective of a project.

TBD

Boundary_Conditions:

 Dependencies

 Legal

 Infrastructure

 Standards

 Certification

 Resources

Structure of requirements at the Project level

(needed for generation of R&S document)

1. Organisation name.
2. Organisation logo.
3. Project title.
4. Overview.
5. Contact information about developers team.
6. Scope (description of future projects results).
 - 6.1. Products – concrete i.e. identifying by name product(s) should be the result of project development (software, hardware, manufactured goods, technology etc.).
 - 6.2. Purpose – how the product(s) will be used.
 - 6.3. Functions – summary of the major functions that product(s) will perform.
 - 6.4. Benefits of product(s) using.
 - 6.5. Possibilities and limitations of products using.
 - 6.6. Product perspective (possibility of future release comparative with other related products).
7. End-users characteristics (educational level, skills, experience, technical expertise, operations).
Operations – set of normal and special actions should be performed by the user.
8. Developers characteristics (educational level, skills, experience, technical expertise, responsibility, rights).
9. Interfaces
 - 9.1. System interfaces – list of system interfaces and its related functionality (describing communications of entities between themselves and environment).
 - 9.2. User interfaces – list of statements describing how the system will appear to the user.
Include logical characteristics of each interface between the systems and its users.
10. Environment characteristics – definition of environment in terms of communication with system via interfaces.
11. Work plan (defined in terms of workpackages and tasks, milestones, deadlines, deliverables) – needs time entities decomposition.
12. Definitions, Acronyms, and Abbreviations (all terms used at R&S development).
13. References (list of all documents and other materials referenced in R&S document).

End-users, developers characteristics, work plan, interfaces, environment definitions can be considered as kinds of specific requirements. It should be transformed from specific requirements to project level at final stage (i.e. at process of generation of R&S document).

Concepts of meta-model are specific characteristics (kind, level, group, category, pattern, priority) of requirements.

Definition of units of measure and timing is one of stages of requirement development (it is similar to entity decomposition). Its choosing and using transform Requirements into Specifications.

Specification, test cases, fault cases, work packages, issues are kinds (descendants) of requirement.

Definitions of the Systems grammar concepts

Divided into base and supplementary concepts

Concept	Definition	Example	Commentary
Entity	Entity that constitutes part of the whole (of a system)	Engine (of a car)	Entity is the base concept of OpenSpecs methodology. 'Base' means that any other notions of OpenSpecs' grammar concern the notion 'Entity'. At program level it is formal frame (class) for reflection of separate notion of modelled subject area
Subject area	Any part of physical, technical etc. reality, which can be the object of research and modelling	Mechanical engineering	In addition to generic meta-model it cannot be excluded that domain specific meta-models have to be developed A subject area defines the specifics of concrete meta-model (structure of repository)
System	1. Set of interacting entities which has a well defined structure and can be considered as a whole 2. A composition of interacting entities, cooperating towards completion of a mission	Car	A system can be composed of several sub-systems. Each system and sub-system is also an entity. Using notion 'system' expediently in case of considering its structure (i.e. interactions between entities). Interactions are expressed by an event-method mechanism and define a behaviour of the system
Requirement	Description of all potentially desired properties of the system being developed, without taking into account a selected architecture or implementation strategy	Protection against fire	Initial requirements are description of system purpose (i.e. target function) expressed in natural language Requirements can be general (concerning every entity in developing system) and local (concerning a specific entity only)
Specification	Requirements expressed as quantifiable and measurable entities	Power of engine should be 120 horse powers	Result of work with OpenSpecs is specifications document, which defines a system ready for implementation
Name	Obligatory attribute of any structural element of entity	Car, Engine	Names have a hierarchical structure reflecting the subordination of entities (for example, Car.Engine). A Name is formed in conformity with the rules of creating identifiers
Purpose (Mission)	Highest level requirement of the system	Purpose of car engine is generating mechanical energy	Purpose reflect qualitative target characteristic of entities Purpose is an attribute of the system and when achieved, the mission can be seen as 'an emerging property' of the system
Property	Reflection of a qualitative or quantitative characteristic of an entity	Engine Power	Qualitative or quantitative characteristics of entities at architectural level implemented by pairs of attributes-values (quality-quantity)
Value	Fixed quantitative characteristic of attribute	100 km per hour	Value of Attribute has own type, initial value and limits. Access to value of attribute is provided by its name
Type	Determines a value and a possible set of operations on it	Integer	Base Types of OpenSpecs will include simple types: Integer, Real, Character, String and Boolean, structural types: Arrays and Records, and mechanism of defining new types by means of reference on base types

Event	An occasion which can take a place in modelled system	Engine start	Event is a result of system property changing
Message	Way of event transferring, implements mechanism of entities interaction	TurnOnLight	Message is used for data transferring between entity and for launching methods of entity Message is a part of interaction protocol and act like synchronization points
Protocol	Form and structure of message mechanism implementation	Message Sequence Chart between driver and engine	Set of rules to be obeyed by the interface. Include the message's content and its sequence in relation to other messages
Method	Concept for function and interaction technical implementation	Engine can have a method PowerRising (NewPower: Integer) to process event 'FuelIncreasing'	Technically behaviour of entity is implemented by means of pairs of event-method. Entity carries out its behaviour by methods launching if they have appropriate event-method conformity
Hierarchy	Relation of subordination between entities	Engine is a structural part of a car	Hierarchy will reflect as spatial and time relations
Status	Notion for expression of a time dimension of system development		It should be considered several stages of entity development: 1. Identification – defining Name of Entity and its Purpose. Description of entities position in the system hierarchy. 2. Atributisation – defining pairs of attributes-values. 3. Behaviour definition – defining pairs of events-methods processing. 4. Implementation of methods of entities and testing.
Project	Set of Tasks for system development	The project to develop a car	All activities related to the development of a system, bounded by specified resources and requirements
Work Package	A coherent set of tasks executed in the process of the project execution	To develop engine of a car	
Task	Well defined, specific element of a Work Package	To define kind of engine	
Work Plan	Timeline or schedule of the tasks to be executed to develop a system		
State	Set of values of system attributes, unambiguously specifying its conditions		
States Space	All possibly System Conditions		
Valid States	Part of general Space of System conditions, which is met by test cases		
Function	Define behaviour of an entity	When wheel blocks, involve ABS	Define transition of entity from one state to another, which belong to Space of Systems States

Behaviour	Defined activity of an entity to implement its mission		
Parallelism	General term for definition of simultaneous actions of coexisting entities		An activity can execute (or continue executing) whenever it has received or send necessary messages
Concurrency	Way of parallelism implementation		Technical method of concurrency implementing: each entity of system is a separate stream of controlling (thread)
Testing	Process of system validity checking in different conditions		Testing is a developer action, checking conformity requirements with test cases Testing assumes changing characteristic parameters of system and checking conformity of behaviour of system to specifications
Test case	Separate requirement for system validity	Power of engine is enough to overcome heights in 30% at speed 60 km per hour	Test case is set by value of critical parameters in different conditions. Test case check validity of system transition from initial to set condition
Failure Case	Identified path of operation that can result in a systems failure		
Refinement	Analysis of non-functional (superfluous) requirements to simplify system structure		Detailed elaboration of requirements
Validation	The process to certify that a completed system satisfies the agreed upon specifications and meets the systems requirements		In time the Original requirements and specifications are dynamical upgrading
Verification	Procedure to test the specifications to proof the implementation of expressed requirements		
Decomposition	Structuring of a system of initial requirements into multiple parts according to different criteria, e.g. subrequirements, specifications, test cases, work packages etc.		The main is the decomposition into entities which links other kinds of decompositions
Interface	Point of interaction between an entity and another entity		
Operator	Human or artificial system interacting with the developing system whose internal behaviour might not be known	The driver of the car	
User	Human or artificial system benefiting from the system without necessarily operating it	Passenger of a car	Party who uses the system under development

Developer	A person participating in a project and actually participating in implementing from design specification to validation testing	The worker of automobile factory	
Environment	System in which the system under development is placed		Environment is realised as a special kind of entity
Dependency area	Part of system(s) on which an entity depends	ABS depends on wheels moving	Dependencies can be indirect
Attribute	An inherent characteristic of an entity		For example, after allocation of requirements to an entity for fulfilment, this entity has a list of fulfilment requirements as an attribute. The entity will also have list of test cases as an attribute to make the requirements meaningful
Problem	Any issue that comes up during the system development process and needs further work to remove or settle the issue		
Responsibility area	Part(s) of a system for which a developer is responsible	The software group shall implement the new algorithm	Responsibility area can be local – what a single developer works with at one time (normally an entity) and group, what a small group of developers is responsible for, (normally a System)
Qualifier	Information, attached to a requirement to define its scope		General : applies to the whole system Normal, applies to part of a system Responsibility, to be fulfilled by the current entity or by one of its support entities Fulfillment to be tested with the current entity

The Definitions of Systems Grammar Terms accordingly to the Britannica dictionary (www.britannica.com)

Term	Meanings used in the System Grammar
Entity	1a: BEING, EXISTENCE ; <i>especially</i> : independent, separate, or self-contained existence b : the existence of a thing as contrasted with its attributes 2 : something that has separate and distinct existence and objective or conceptual reality
Interaction	1 : mutual or reciprocal action or influence
System	1 : a regularly interacting or interdependent group of items forming a unified whole <a number <i>system</i> >: as a (1) : a group of interacting bodies under the influence of related forces <a gravitational <i>system</i> > (2) : an assemblage of substances that is in or tends to equilibrium <a thermodynamic <i>system</i> > b (1) : a group of body organs that together perform one or more vital functions <the digestive <i>system</i> > (2) : the body considered as a functional unit c : a group of related natural objects or forces <a river <i>system</i> > d : a group of devices or artificial objects or an organization forming a network especially for distributing something or serving a common purpose <a telephone <i>system</i> > <a heating <i>system</i> > <a highway <i>system</i> > <a data processing <i>system</i> >
Property	1 a : a quality or trait belonging and especially peculiar to an individual or thing b : an effect that an object has on another object or on the senses c : VIRTUE 3 d : an attribute common to all members of a class
Attribute	1 : an inherent characteristic; <i>also</i> : an accidental quality 2 : an object closely associated with or belonging to a specific person, thing, or office <a scepter is the <i>attribute</i> of power>; <i>especially</i> : such an object used for identification in painting or sculpture 3 : a word ascribing a quality; <i>especially</i> : ADJECTIVE
Interface	1 : a surface forming a common boundary of two bodies, spaces, or phases <an oil-water <i>interface</i> > 2 a : the place at which independent and often unrelated systems meet and act on or communicate with each other <the man-machine <i>interface</i> > b : the means by which interaction or communication is achieved at an interface
Action	5 b : the accomplishment of a thing usually over a period of time, in stages, or with the possibility of repetition c plural : BEHAVIOR, CONDUCT 7 a : an operating mechanism b : the manner in which a mechanism or instrument operates
Behavior	1 a : the manner of conducting oneself b : anything that an organism does involving action and response to stimulation c : the response of an individual, group, or species to its environment 2 : the way in which someone behaves; <i>also</i> : an instance of such behavior 3 : the way in which something functions or operates
Function	2 : the action for which a person or thing is specially fitted or used or for which a thing exists : PURPOSE 3 : any of a group of related actions contributing to a larger action; <i>especially</i> : the normal and specific contribution of a bodily part to the economy of a living organism 5 a : a mathematical correspondence that assigns exactly one element of one set to each element of the same or another set b : a variable (as a quality, trait, or measurement) that depends on and varies with another <height is a <i>function</i> of age>; <i>also</i> : RESULT <illnesses that are a <i>function</i> of stress> 7 : a computer subroutine; <i>specifically</i> : one that performs a calculation with

	variables provided by a program and supplies the program with a single result
Operation	<p>1 : performance of a practical work or of something involving the practical application of principles or processes</p> <p>2 a : an exertion of power or influence <the <i>operation</i> of a drug> b : the quality or state of being functional or operative <the plant is now in <i>operation</i>> c : a method or manner of functioning <a machine of very simple <i>operation</i>></p> <p>5 : any of various mathematical or logical processes (as addition) of deriving one entity from others according to a rule</p> <p>8 : a single step performed by a computer in the execution of a program</p>
Purpose	<p>1 a : something set up as an object or end to be attained : INTENTION b : RESOLUTION, DETERMINATION</p> <p>2 : a subject under discussion or an action in course of execution</p>
Event	<p>1 a <i>archaic</i> : OUTCOME b : the final outcome or determination of a legal action c : a postulated outcome, condition, or eventuality <in the <i>event</i> that I am not there, call the house></p> <p>2 a : something that happens : OCCURRENCE b : a noteworthy happening</p>
Message	<p>1 : a communication in writing, in speech, or by signals</p> <p>2 : a messenger's errand or function</p> <p>3 : an underlying theme or idea</p>
Protocol	<p>1 : an original draft, minute, or record of a document or transaction</p> <p>3 b : a set of conventions governing the treatment and especially the formatting of data in an electronic communications system</p> <p>4 : a detailed plan of a scientific or medical experiment, treatment, or procedure</p>
Port	<p>5 : a hardware interface by which a computer communicates with another device or system</p>
Specification	<p>1 : the act or process of specifying</p> <p>2 a : a detailed precise presentation of something or of a plan or proposal for something -- usually used in plural b : a statement of legal particulars (as of charges or of contract terms); <i>also</i> : a single item of such statement c : a written description of an invention for which a patent is sought</p>
Requirement	<p>1: something required: a : something wanted or needed : NECESSITY <production was not sufficient to satisfy military <i>requirements</i>> b : something essential to the existence or occurrence of something else : CONDITION <failed to meet the school's <i>requirements</i> for graduation></p>
Mission	<p>1 <i>obsolete</i> : the act or an instance of sending</p> <p>2 a : a ministry commissioned by a religious organization to propagate its faith or carry on humanitarian work b : assignment to or work in a field of missionary enterprise c (1) : a mission establishment (2) : a local church or parish dependent on a larger religious organization for direction or financial support d plural : organized missionary work e : a course of sermons and services given to convert the unchurched or quicken Christian faith</p> <p>3 : a body of persons sent to perform a service or carry on an activity: as a : a group sent to a foreign country to conduct diplomatic or political negotiations b : a permanent embassy or legation c : a team of specialists or cultural leaders sent to a foreign country</p> <p>4 a : a specific task with which a person or a group is charged b (1) : a definite military, naval, or aerospace task <a bombing <i>mission</i>> <a space <i>mission</i>> (2) : a flight operation of an aircraft or spacecraft in the performance of a mission <a</p>



<i>mission to Mars></i> 5 : CALLING, VOCATION
--