CLAIMS

1.- A development system (10) for developing a railway signalization program (11), pursuant to standards CENELEC EN 50128 and CEI 62279, the railway signalization program (11) being adapted to be carried out by a railway signalization system, the development system (10) comprising:

- a text language programming module (12) for programming, by at least one user, of at least one computer operator (14, 15, 16, 17), the or each computer operator (14, 15, 16, 17) processing at least one predetermined type of incoming data compatible with the computer operator (14, 15, 16, 17), and producing at least one predetermined type of outgoing data compatible with the computer operator (14, 15, 16, 17);

- a memory (18) for the storage of the or each computer operator (14, 15, 16, 17) that is programmed with the text language programming module (12) ;

- a graphic language programming module storing graphic language programming software converting the or each computer operator (14, 15, 16, 17) stored in said memory (18) in a form of a functional graphic element (52, 53, 54, 55, 56, 57), the or each functional graphic element (52, 53, 54, 55, 56, 57) having, for each predetermined type of incoming data, a respective input for supplying the computer operator (14, 15, 16, 17) with at least one incoming datum of the predetermined type, and, for each predetermined type of outgoing data, a respective output for providing, by the computer operator (14, 15, 16, 17), at least one outgoing datum of the predetermined type;

- a visual environment (50) for the presentation of the or each functional graphic element (52, 53, 54, 55, 56, 57) provided by the graphic language programming module, , the visual environment (50) being suitable for allowing the construction, by at least one user, of a visual representation (22) of the railway signalization program (11) from the or each functional graphic element (52, 53, 54, 55, 56, 57) provided by the graphic language programming module;

- an implementation module (24) able to implement the railway signalization program (11) based on the visual representation (22), the implementation module (24) comprising a compiler to compile the visual representation (22) in a language executable by the railway signalization system, or an interpreter to execute the railway signalization program (11) directly from the visual representation (22).

2.- The development system (10) according to claim 1, wherein the visual environment (50) makes it possible to duplicate the or each functional graphic element (52, 53, 54, 55, 56, 57), so as to form a duplicated graphic element.

3.- The development system (10) according to claim 1 or 2, wherein the visual environment (50) comprises at least one graphic stream element (100, ..., 121) for connecting the inputs and the outputs of the or each functional graphic element (52, 53, 54, 55, 56, 57) so as to build the visual representation (22).

4.- The development system (10) according to claim 3, wherein the visual environment (50) is suitable for preventing the connection of inputs corresponding to a predetermined type of data to outputs corresponding to a predetermined type of data different from said predetermined type and which does not encompass or constitute a subtype of said predetermined type.

5.- The development system (10) according to claim 3 or 4, wherein each computer operator (14, 15, 16, 17) has, for each type of incoming and outgoing data that it is configured to process, respectively to produce, a cardinality of said data type defining the number of data of said type that the computer operator (14, 15, 16, 17) is suitable for processing, respectively for producing, upon each instance of said computer operator (14, 15, 16, 17).

6.- The development system (10) according to claim 5, wherein at least one computer operator (14, 15, 16, 17) has a fixed cardinality for at least one type of incoming or outgoing data that it is configured to process, respectively to produce, and the visual environment (50) is suitable for preventing the connection, to the or each input and/or to the or each output with which a fixed cardinality is associated, of a number of graphic stream elements (100, ..., 121) different from said cardinality.

7.- The development system (10) according to claim 5 or 6, wherein at least one computer operator (14, 15, 16, 17) has a variable cardinality for at least one type of incoming or outgoing data that it is configured to process, respectively to produce.

8.- The development system (10) according to any one of claims 3 to 7, wherein the visual environment (50) comprises at least one aggregator graphic element (80) for aggregating several basic-types data into at least one complex-type datum, the visual environment (50) also comprising at least one graphic stream element (100, ..., 121) for connecting at least one input and/or one output of the aggregator graphic element (80) to at least one output, respectively to at least one input of a functional graphic element (52, 53, 54, 55, 56, 57).

9.- The development system (10) according to claim 8, wherein the aggregator graphic element (80) has, for each basic type, a cardinality of said basic type defining the number of data of said basic type that the aggregator graphic element (80) is suitable for

aggregating within the complex-type datum, the cardinality associated with at least one of said basic types being variable.

10.- The development system (10) according to any one of claims 3 to 9, wherein the visual environment (50) comprises at least one decomposer graphic element (82) for the decomposition of at least one complex-type datum into several basic-types data, the visual environment (50) also comprising at least one graphic stream element (100, ..., 121) for connecting an input and/or at least one output of the decomposer graphic element (82) to at least one output, respectively to at least one input, of a functional graphic element (52, 53, 54, 55, 56, 57).

11.- The development system (10) according to claim 1, wherein the text language programming module (12) comprises a man-machine interface for programming, by at least one user, the or each computer operator (14, 15, 16, 17).

12.- A development method (200) for developing a railway signalization program (11) using a development system (10) according to any one of the preceding claims, characterized in that it comprises the following steps:

- programming (210), by a first user, using the text language programming module (12) in text language (12) of the development system (10), of a plurality of computer operators (14, 15, 16, 17), each computer operator (14, 15, 16, 17) processing at least one predetermined type of incoming data compatible with the computer operator (14, 15, 16, 17), and producing at least one predetermined type of outgoing data compatible with the computer operator (14, 15, 16, 17), wherein said programming is carried out pursuant to standards CENELEC EN 50128 and CEI 62279,
- storing (220) each programmed computer operator (14, 15, 16, 17) in the memory (18),
- converting each computer operator (14, 15, 16, 17) stored in said memory (18) in a form of a functional graphic element (52, 53, 54, 55, 56, 57) using the graphic language programming module of the development system (10), the or each functional graphic element (52, 53, 54, 55, 56, 57) having, for each predetermined type of incoming data, a respective input for supplying the computer operator (14, 15, 16, 17) with at least one incoming datum of the predetermined type, and, for each predetermined type of outgoing data, a respective output for providing, by the computer operator (14, 15, 16, 17), at least one outgoing datum of the predetermined type,

- presenting (230) each functional graphic element (52, 53, 54, 55, 56, 57) in the visual environment (50) of the development system (10),
- building (240) a visual representation (22) of the railway signalization program (11), by a second user, in the visual environment (50), from functional graphic elements (52, 53, 54, 55, 56, 57), and
- implementing (250) the railway signalization program (11) from the visual representation (22), via the implementation module (24), wherein the implementation module (24) compiles the visual representation (22) in a language executable by the railway signalization system, or executes the railway signalization program directly from the visual representation (22).

13.- The development method (200) according to claim 12, wherein the or each computer operator (14, 15, 16, 17) is configured to process at least one predetermined type of incoming data and to produce at least one predetermined type of outgoing data, and the or each functional graphic element (52, 53, 54, 55, 56, 57) converted from the computer operator has a respective input for supplying the computer operator (14, 15, 16, 17) with at least one incoming datum of said predetermined type, and, for the or each predetermined type, a respective output for the provision, by the computer operator (14, 15, 16, 17), of at least one outgoing datum of said predetermined type.

14.- The development method (200) according to claim 12 or 13, wherein the step for building the visual representation (240) comprises a sub-step (243) of the second user connecting, in the visual environment (50), outputs of several low-level functional graphic elements to the inputs of an aggregator graphic element (80), and the presentation (244), by the visual environment (50), of a representation of a complex-type data stream representing a complex-type data stream resulting from the aggregation of output data produced by the computer operator(s) (14) represented by said low-level functional graphic elements.

15.- The development method (200) according to any one of claims 12 to 14, wherein the step of building the visual representation (240) comprises a sub-step (246) of the second user connecting, in the visual environment (50), an output of a high-level functional graphic element to an input of a decomposer graphic element (82), and the presentation, by the visual environment (50), of representations of basic-type data streams representing basic-type data streams resulting from the decomposition of an outgoing datum produced by the computer operator represented by the high-level functional graphic element.