

fe.



Industria y Comercio
SUPERINTENDENCIA

**SOLICITUD
PATENTE DE INVENCION**

E-XPEDIENTE N°

05-1042

TITULO: GENERACIÓN AUTIMATIZADA DE
CÓDIGO DE SIMULACIÓN DE PATRONES DE
INTERCAMBIO DE MENSAJES

CLASIFICACION INTERNACIONAL: G06 F 17/60

SOLICITANTE: MICROSOFT CORPORATION

DOMICILIO: REDMOND, WASHINGTON, ESTADOS
UNIDOS DE AMERICA

APODERADO: LUZ HELENA ADARVE GOMEZ

BOGOTÁ, D.C. _____

PETITORIO

7

Industria y Comercio
SUPERINTENDENCIA

SUPERINDUSTRIA Y COMERCIO
Radicación : 05801042 00000000 Folios: 115
Fecha (AMB): 2003-01-07 12:14:23
Trámite : 002 PATENTES B 1 REGISTRAR/ 411 PRESENTAR
Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

FORMULARIO UNICO DE SOLICITUD DE PATENTE 2000-3

1 SOLICITUD DE:			
☒ Patente de Invención		☐ Patente de Modelo de Utilidad	
2 SOLICITANTE (71)	Nombre: MICROSOFT CORPORATION Dirección: One Microsoft Way Redmond, Washington 98052, Estados Unidos de America Teléfono: 3123800 Fax 3122410 E-Mail: general@cardenasvcardenas.com Lugar de Constitución: Estado de Washington, Estados Unidos de América Domicilio: REDMOND, WASHINGTON, ESTADOS UNIDOS DE AMERICA		IDENTIFICACION C.C. ☐ NIT ☐ C.E. ☐ Otro ☒ Número _____
3 REPRESENTANTE O APODERADO (74)	Nombre: LUZ HELENA ADARVE GOMEZ Dirección: Carrera 7 No 71-52 Torre B Piso 9 Teléfono: 3123800 Fax 3122410 E-Mail: general@cardenasvcardenas.com		IDENTIFICACION C.C. ☒ NIT ☐ C.E. ☐ Otro ☐ Número 41.575.435 BTA TP 14.884 C.S.J.
4 INVENTOR(ES) (72)	Nombre: - George P. Copeland - Jason Allen - Luis Felipe Cabrera Dirección: - One Microsoft Way Redmond, Washington 98052, Estados Unidos de America - One Microsoft Way Redmond, Washington 98052, Estados Unidos de America - One Microsoft Way Redmond, Washington 98052, Estados Unidos de America Teléfono: 3123800 Fax 3123800 E-Mail: general@cardenasvcardenas.com Domicilio: - WASHINGTON, ESTADOS UNIDOS DE AMERICA - WASHINGTON, ESTADOS UNIDOS DE AMERICA - WASHINGTON, ESTADOS UNIDOS DE AMERICA		IDENTIFICACION C.C. ☐ NIT ☐ C.E. ☐ Otro ☒ Número _____
5 Título(54): GENERACIÓN AUTOMATIZADA DE CÓDIGO DE SIMULACIÓN DE PATRONES DE INTERCAMBIO DE MENSAJES			
6 Prioridad		(33) País de Origen Estados Unidos de América	(32) Fecha Enero 23, 2004
SI ☒ NO ☐		(31) N° de Solicitud 10/783,311	
7 Para publicar a partir de la fecha de la presente solicitud a los:			
☐ 6 meses ☐ 12 meses ☐ 18 meses ☐ Otro			

CLASIFICACIÓN INTERNACIONAL
G06F 17/60

Comprobante de Pago No.:

05-1,745 / 05,746

Fecha

Enero 8, 2005

ANEXOS

- ☒ Comprobante de pago de la tasa de presentación de la solicitud. (05-1,745)
- ☐ Comprobante de pago de la tasa por concepto de excedente de palabras en la publicación.
- ☒ Comprobante de pago de la tasa por concepto de reivindicación de prioridad. (05-1,746).
- ☒ Poderes, si fuere el caso. (Protocolo No 16.069)
- ☒ Certificado de existencia y representación legal cuando el solicitante sea persona jurídica. Protocolo No 16.069)
- ☐ Copia certificada de la primera solicitud, si se reivindica prioridad.
- ☐ Traducción simple de la primera solicitud, si se reivindica prioridad.
- ☐ Documento de cesión del inventor al solicitante o a su causante.
- ☒ Descripción de la invención.
- ☒ Una o más reivindicaciones.
- ☒ Dibujos y/o planos necesarios.
- ☐ De ser el caso, copia del contrato de acceso.
- ☐ De ser el caso, documento que acredite la licencia o autorización de uso de conocimientos tradicionales de la comunidad de las comunidades indígenas.
- ☐ De ser el caso, certificado de depósito del material biológico.
- ☒ Arte final 12x12 y 6x6 cm.
- ☒ Resumen y extracto para publicación.

10

NOMBRE: LUZ HELENA ADARVE GOMEZ

FIRMA:


C.C. 41.575.435 BTA TP. 14.384
C.SJ.

SUPERINDUSTRIA Y COMERCIO
Radacion : 05001042 00000000 Folios: 115
Fecha (AMD): 2005-01-07 12:14:23
Tramite : 008 PATENTES D 1 REGISTRO/ 411 PRESENTAC
Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

NIT : 800176089-2

RECIBO OFICIAL DE CAJA : 05 - 1,745
FECHA : ENERO 6 DE 2005

CONSIGNACION

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I.	CONSIGNACION	BANCO POPULAR	050-00110-6	37957146	06/01/2005	8,957,000.00

CONCEPTO

CANT. CONTABILITICO	CONCEPTO	TOTAL CONCEPTO
1 50005-01-01 SOLICITUDES	1 TRAMITES DE SOL. DE PATENTE DE IN	443,000.00
	TOTAL :	\$ 443,000.00

SOM: CUATROCIENTOS CUARENTA Y TRES MIL PESOS

RESPONSABLE : _____

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____

GENERACION AUTOMATIZADA DE COOZGO
DE SIMULACION DE PATRONES DE
INTERCAMBIO DE MENSAJES.

CARDENAS & CARDENAS
ABOGADOS

SUPERINDUSTRIA Y COMERCIO
 Radicacion : 05001042 00000000 Folios: 115
 Fecha (MM): 2005-01-07 12:14:23
 Trámite : 002 PATENTES D 1 REGISTRAR/ 411 PRESENTAR
 Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

SUPERINTENDENCIA DE INDUSTRIA Y COMERCIO

MIT : 800176089-2

RECIBO OFICIAL DE CAJA : 05 - 1,746
 FECHA : ENERO 6 DE 2005

***** CONSIGNACION *****

DEPOSITANTE	TIPO PAGO	BANCO	CUENTA	No. PAGO	FECHA PGO	Vr. PAGO
CARDENAS Y CARDENAS P.I.	CONSIGNACION	BANCO POPULAR	050-00110-6	37957146	06/01/2005	8,957,000.00

***** CONCEPTO *****

CANT.	RENTISTICO	CONCEPTO	TOTAL CONCEPTO
1	50005-01-01 SOLICITUDES	82 INVOCACION DE UNA PRIORIDAD EN NU	124,000.00
		TOTAL :	\$ 124,000.00

SON: CIENTO VEINTICUATRO MIL PESOS

RESPONSABLE : _____

RECIBO DE CAJA APLICADO AL EXPEDIENTE No. _____

GENERACION AUTOMATIZADA DE
 CODIGO DE SIMULACION DE PATRONES
 DE INTERCAMBIO DE MENSAJES.

GARDENAS & GARDENAS
ABOGADOS

UNITED STATES PATENT APPLICATION

of

Luis Felipe Cabrera,

George P. Copeland

and

Jason Allen

for

**AUTOMATED GENERATION OF
MESSAGE EXCHANGE PATTERN SIMULATION CODE**

6

**AUTOMATED GENERATION OF
MESSAGE EXCHANGE PATTERN SIMULATION CODE**

BACKGROUND OF THE INVENTION

1. The Field of the Invention

[0001] The present invention relates to networking technologies; and more specifically, to the automated generation of code that simulates a message exchange pattern.

2. Background and Related Art

[0002] Computing technology has transformed the way we work and play. Computing systems now take a wide variety of forms including desktop computers, laptop computers, tablet PCs, Personal Digital Assistants (PDAs), and the like. Even household devices (such as refrigerators, ovens, sewing machines, security systems, and the like) have varying levels of processing capability and thus may be considered computing systems.

[0003] Much of the functionality provided by computing systems relies on the computing system being networked with (and able to communicate messages with) other computing systems. In order for the two communicating computing systems to accomplish a particular task, a message transaction involving a pattern of message exchanges may be needed. Each computing system involved in the message transaction has an application (hereinafter also referred to as a "message exchange pattern application") that assists in processing messages received as part of the transaction that follows the message exchange pattern, and transmitting other messages that conform with the message exchange pattern.

[0004] In order to test a message exchange pattern application, the message exchange pattern application may be installed on different computing systems and then tested.

7

However, in many cases, the message exchange pattern application on one computing system may be significantly different than the message exchange pattern application on another computing system, even though the two message exchange pattern applications are capable of interacting with each other using the message exchange pattern. The tester might only have access to one of the message exchange pattern applications. Furthermore, the owner of the other message exchange pattern application may not be willing or able to cooperatively test the interaction of the two message exchange pattern applications. Accordingly, it is not always feasible to test the message exchange pattern application in an actual network environment in which it will be implemented and deployed.

[0005] What would be advantageous is a mechanism for testing message exchange pattern applications without having to test the application in the actual network environment in which it would ultimately be employed. It would furthermore be advantageous if this testing may be accomplished without requiring the authoring of additional testing software each time a message exchange pattern application is to be tested.

7

BRIEF SUMMARY OF THE INVENTION

[0006] The foregoing problems with the prior state of the art are overcome by the principles of the present invention, which are directed towards mechanisms for automatically generating code that tests capabilities of a test computing system to simulate a message exchange pattern.

[0007] The code generation computing system uses a message exchange pattern definition to generate the simulation code. The message exchange pattern definition includes the following for each of the transition states in the message exchange pattern: 1) an indication of which computing system may transmit (and/or receive) each valid message given the state, and 2) a state transition description resulting from transmitting or receiving a valid message.

[0008] For each state in which the message exchange pattern definition allows valid messages to be transmitted, code is generated for that state that at least simulates the transmission of a valid transmission message. For each state in which the message exchange pattern definition allows valid messages to be received, code is generated for that state that at least simulates the receipt of a valid receipt message. If the transmission or receipt of the message causes a state transition to occur, code is generated that causes the appropriate state transition in the message exchange pattern. The code generation computing system may also generate code that simulates the transmission or receipt of invalid messages given a particular state as well. Probabilities may be assigned to certain message(s) (whether valid or invalid) being transmitted or received. The message exchange pattern may be simulated a number of times to simulate many or all possible state transition paths through the message exchange pattern. This enables the simulation to cover even potentially all of the possible cases that will be encountered in a real deployment.

9

[0009] Accordingly, the simulation code is automatically generated using the message exchange pattern definition, thereby making it easier to generate simulation code. Furthermore, the simulation may be internal to a single computing system since message transmission and receipt may be simulated without communicating with another computing system. Accordingly, there is no need to use network resources. Alternatively, even if the simulation is not internal to the single computing system, the simulation allows for communication within the network of a single organization. Accordingly, there is no need to rely on the cooperation of another organization in order to test the message exchange pattern application.

[0010] Additional features and advantages of the invention will be set forth in the description that follows, and in part will be obvious from the description, or may be learned by the practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the present invention will become more fully apparent from the following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

9

BRIEF DESCRIPTION OF THE DRAWINGS

[0011] In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail through the use of the accompanying drawings in which:

[0012] Figure 1 illustrates a suitable computing system that may implement features of the present invention;

[0013] Figure 2 illustrates a network environment in which two computing systems implement a message exchange pattern, each using a message exchange pattern application;

[0014] Figure 3 illustrates a code generation computing system then automatically generates code that internally simulates a message exchange pattern on a test computing system;

[0015] Figure 4 illustrates a test computing system internally simulating the message exchange pattern using simulation code generated by the code generation computing system;

[0016] Figure 5 illustrates a state transition tree that may be used to track progress in an arbitrary message exchange pattern; and

[0017] Figure 6 illustrates a flowchart of a method for the code generation computing system to automatically generate simulation code that internally simulates a message exchange pattern.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0018] The principles of the present invention relate to mechanisms for automatically generating code that tests capabilities of a test computing system to simulate a message exchange pattern. The code generation computing system uses a message exchange pattern definition as a basis for automatically generating the simulation code. For each state in which the message exchange pattern definition allows valid messages to be transmitted, code is generated for that state that at least simulates the transmission of a valid transmission message. For each state in which the message exchange pattern definition allows valid messages to be received, code is generated for that state that simulates the receipt of a valid receipt message. If the transmission or receipt of the message causes a state transition to occur, code is generated that causes the appropriate state transition in the message exchange pattern.

[0019] Turning to the drawings, wherein like reference numerals refer to like elements, the invention is illustrated as being implemented in a suitable computing environment. The following description is based on illustrated embodiments of the invention and should not be taken as limiting the invention with regard to alternative embodiments that are not explicitly described herein.

[0020] In the description that follows, the invention is described with reference to acts and symbolic representations of operations that are performed by one or more computers, unless indicated otherwise. As such, it will be understood that such acts and operations, which are at times referred to as being computer-executed, include the manipulation by the processing unit of the computer of electrical signals representing data in a structured form. This manipulation transforms the data or maintains them at locations in the memory system of the computer, which reconfigures or otherwise alters the operation of the computer in a

n

manner well understood by those skilled in the art. The data structures where data are maintained are physical locations of the memory that have particular properties defined by the format of the data. However, while the invention is being described in the foregoing context, it is not meant to be limiting as those of skill in the art will appreciate that several of the acts and operations described hereinafter may also be implemented in hardware. Figure 1 shows a schematic diagram of an example computer architecture usable for these devices.

[0021] For descriptive purposes, the architecture portrayed is only one example of a suitable environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing systems be interpreted as having any dependency or requirement relating to anyone or combination of components illustrated in Figure 1.

[0022] The invention is operational with numerous other general-purpose or special-purpose computing or communications environments or configurations. Examples of well known computing systems, environments, and configurations suitable for use with the invention include, but are not limited to, mobile telephones, pocket computers, personal computers, servers, multiprocessor systems, microprocessor-based systems, minicomputers, mainframe computers, and distributed computing environments that include any of the above systems or devices.

[0023] In its most basic configuration, a computing system 100 typically includes at least one processing unit 102 and memory 104. The memory 104 may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.), or some combination of the two. This most basic configuration is illustrated in Figure 1 by the dashed line 106.

[0024] The storage media devices may have additional features and functionality. For example, they may include additional storage (removable and non-removable) including, but

not limited to, PCMCIA cards, magnetic and optical disks, and magnetic tape. Such additional storage is illustrated in Figure 1 by removable storage 108 and non-removable storage 110. Computer-storage media include volatile and non-volatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, data structures, program modules, or other data. Memory 104, removable storage 108, and non-removable storage 110 are all examples of computer-storage media. Computer-storage media include, but are not limited to, RAM, ROM, EEPROM, flash memory, other memory technology, CD-ROM, digital versatile disks, other optical storage, magnetic cassettes, magnetic tape, magnetic disk storage, other magnetic storage devices, and any other media that can be used to store the desired information and that can be accessed by the computing system.

[0025] As used herein, the term "module" or "component" can refer to software objects or routines that execute on the computing system. The different components, modules, engines, and services described herein may be implemented as objects or processes that execute on the computing system (e.g., as separate threads). While the system and methods described herein are preferably implemented in software, implementations in software and hardware or hardware are also possible and contemplated.

[0026] Computing system 100 may also contain communication channels 112 that allow the host to communicate with other systems and devices over a network 120. Communication channels 112 are examples of communications media. Communications media typically embody computer-readable instructions, data structures, program modules, or other data in a modulated data signal such as a carrier wave or other transport mechanism and include any information-delivery media. By way of example, and not limitation, communications media include wired media, such as wired networks and direct-wired

connections, and wireless media such as acoustic, radio, infrared, and other wireless media. The term computer-readable media as used herein includes both storage media and communications media.

[0027] The computing system 100 may also have input components 114 such as a keyboard, mouse, pen, a voice-input component, a touch-input device, and so forth. Output components 116 include screen displays, speakers, printer, etc., and rendering modules (often called "adapters") for driving them. The computing system 100 has a power supply 118. All these components are well known in the art and need not be discussed at length here.

[0028] Figure 2 illustrates a network environment 200 in which two computing systems (referred to as "implementation computing systems" in Figure 2) communicate to accomplish a particular task. In particular, the implementation computing system 201 communicates with the implementation computing system 202. In order to accomplish the particular task, the computing systems 201 and 202 exchange messages in a particular pattern of exchange 230. The particular pattern of message exchange defines which computing system is to send which message at any given point in the message exchange. The message exchange pattern depends on the task to be accomplished, and the protocols used to exchange messages. The principles of the present invention are not limited to any given structure for the message exchange pattern.

[0029] Each implementation computing system has a message exchange pattern application that engages in the message exchange pattern. For example, implementation computing system 201 executes message exchange pattern application 210, while implementation computing system 202 executes message exchange pattern application 220. The implementation computing systems 201 and 202 may be structured as described above

for the computing system although this is not required. In this description and in the claims, a "computing system" is defined as any device or system that has the capability to process electronic messages.

[0030] The messages within the message exchange pattern need not all be generated and consumed by the same components in the computing system. For example, some message may be protocol level messages to allow for proper coordination at the protocol-level between the two computing systems. Other messages may be application-level message to allow for the exchange of data needed to accomplish the desired task. In one embodiment, the protocol level messages in the message exchange pattern are governed by the Web Services Coordination (WS-Coordination) and WS-Transaction specifications, which are known to those of ordinary skill in the art.

[0031] Figure 2 illustrates the environment in which the message exchange pattern applications will ultimately operate. However, it is desirable to test the message exchange pattern application to ensure proper operation prior to shipping the application. It often takes considerable time to test applications since often new code needs to be drafted in order to perform simulation. Furthermore, the testing of message exchange pattern applications is particularly difficult since the operations are by nature distributed, and often distributed across boundaries of trust. For example, those authoring the message exchange pattern application 210 may not have access to the message exchange pattern application 220 that is to run on the other computing system. Furthermore, the cooperation of the owner of the message exchange pattern application 220 may be difficult to acquire for testing purposes.

[0032] Accordingly, the principles of the present invention provide a mechanism whereby code is automatically generated that allows a test computing system to simulate the conduct of a message exchange pattern without requiring extensive human authoring of

code, and without requiring the testing to occur with the cooperation of another organization or even perhaps in any network environment at all. Figure 3 illustrates an environment in which such automated code generation may occur.

[0033] Referring to Figure 3, a code generation computing system 301 accesses a message exchange pattern definition 302. A simulation code generation module 303 uses the message exchange pattern definition 303 to generate simulation code 304. Referring to Figure 4, this simulation code 304 may then be used by a test computing system 401 by a simulator 402 to test the message exchange pattern definition. More regarding this operation will be described further below after having first described the message exchange pattern definition 303 in further detail.

[0034] In the message exchange pattern, there are a number of states. The message exchange pattern definition defines these states. For each state, there will be zero or more valid message types that may be transmitted, and zero or more valid message types that may be received. With the receipt or transmission of a valid message type, the message exchange pattern may transition to a different state. The message exchange pattern indicates the valid message types for each state, and indicates what state transitions should occur given a certain valid message type. The message exchange pattern definition may be in any format. However, in order to interpret the message exchange pattern in order to automatically generate code, it is advantageous if the message exchange pattern definition is in a format that may be more easily parsed. An example of such a format is defined by the Web Services Description Language (WSDL) standard.

[0035] The message exchange pattern definition may also include, for example, probability chances that a certain valid message will be received or transmitted given a certain state. The message exchange pattern definition may also impose timing policies that

17

should be applied given a certain state. For example, timing policies may indicate that a previous request should be cancelled if no response is received within a certain time period.

[0036] The message exchange pattern definition may be better understood by reducing the message exchange pattern definition to a state transition tree. A state transition tree may be constructed in memory during the code generation process although this need not be the case. The message exchange pattern state transition tree includes nodes for each state in the message exchange pattern. For each node, there is provided a listing of valid message types and which computing system may send which types of messages. For each valid message type, there is also an identification of which state to transition to when receiving (or transmitting as the case may be) the specified message.

[0037] The structure of the tree depends entirely on the message exchange pattern, which in turn depends on the task to be performed. Illustrating all possible state transition trees would be impossible due to the endless variety of message exchange patterns. Listing many would obscure the principles of the present invention. However, one message exchange pattern is described here for illustrative purposes only. Those of ordinary skill in the art will appreciate (after having read this description) that the principles of the present invention apply to any message exchange pattern.

[0038] That said, Figure 5 illustrates an example message exchange pattern state transition tree 500. The state transition tree 500 includes a number of states 501 through 507. The transmission of messages from one of the computing systems results in state transitions represented by solid arrows. On the other hand, the transmission of messages from the other of the computing systems results in state transitions represented by dashed arrows. For now, ignore the arrows 541 and 542 as well as the percentage markings, which items will be explained further below with respect to Figure 6. Referring to Figure 5, one

17

computing system transmits messages that result in state transitions 511 through 514, while the other computing system transmits messages that result in state transitions 521 through 527. Each arrow is associated with a particular message type or group of message types that are valid for that transition.

[0039] If WS-Coordination is being used, the implementation computing system 201 may first send a coordination context message to the implementation computing system 202. The implementation computing system 202 then may send a registration message to the implementation computing system 201. The implementation computing system 201 then sends a registration receipt message to the implementation computing system 202. At this stage, the implementation computing system 201 takes on the role of "coordinator" and the implementation computing system 202 takes on the role of "participant", and the message exchange is in the active state 501. The solid lines in Figure 5 represent state transitions caused by the coordinator (i.e., the implementation computing system 201) transmitting messages, while the dashed lines in Figure 5 represent state transitions caused by the participant (i.e., the implementation computing system 202) transmitting messages.

[0040] From the active state 501, the implementation computing system 201 may transmit a cancel message to the implementation computing system 202. This represents that the implementation computing system 201 has chosen to cancel operations previously requested, and result in a transition from the active state 501 to the canceling state 502 as represented by the solid arrow 511. More generally, the transmitting computing system recognizes state transitions as soon as the message associated with the transition is transmitted. The receiving computing system recognizes state transitions as soon as the message associated with the transition is received. Accordingly, there may be some transient differences in the tracked progress of the state transition chart as recognized by the

implementation computing system 201, and the tracked progress of the state transition chart as recognized by the implementation computing system 202. In this specific transition 511, the implementation computing system 201 recognizes this transition 511 as soon as it transmits the cancel message. The implementation computing system 202 recognizes the transition 511 as soon as it receives the cancel message.

[0041] Alternatively, while in the active state 501, the implementation computing system 201 may receive an exited message from the implementation computing system 202. This informs the implementation computing system 201 that the implementation computing system 202 will no longer participate in the message exchange pattern. Accordingly, the transmission of the exited message from the implementation computing system 202 to the implementation computing system 201 results in a transition 522 from the active state 501 to an ended state 505.

[0042] Alternatively, while in the active state 501, the implementation computing system 201 may receive a completed message from the implementation computing system 202. This informs the implementation computing system 201 that the implementation computing system 202 has completed processing related to a previous request. Accordingly, the transmission of the completed message from the implementation computing system 202 to the implementation computing system 201 results in a transition 523 from the active state 501 to a completed state 503.

[0043] Alternatively, while in the active state 501, the implementation computing system 201 may receive a faulted message from the implementation computing system 202. This informs the implementation computing system 201 that the implementation computing system 202 has failed from the active state 501. Accordingly, the transmission of the faulted

message from the implementation computing system 202 to the implementation computing system 201 results in a transition 526 from the active state 501 to a faulted state 507.

[0044] While in the canceling state 502, the implementation computing system 201 may receive a canceled message from the implementation computing system 202. This informs the implementation computing system 201 that the implementation computing system 202 acknowledges that the previously requested operation has been cancelled at the request of the implementation computing system 201. Accordingly, the transmission of the canceled message from the implementation computing system 202 to the implementation computing system 201 results in a transition 521 from the canceling state 502 to the ended state 505.

[0045] While in the completed state 503, the implementation computing system 201 may transmit a close message to the implementation computing system 202. This informs the implementation computing system 202 that the previous request was successful. Accordingly, the transmission of the close message from the implementation computing system 201 to the implementation computing system 202 results in a transition 512 from the completed state 503 to the closing state 504.

[0046] While in the closing state 504, the implementation computing system 201 may receive a closed message from the implementation computing system 202. This informs the implementation computing system 201 that the implementation computing system 202 has finalized the operation successfully. Accordingly, the transmission of the closed message from the implementation computing system 202 to the implementation computing system 201 results in a transition 524 from the closing state 504 to the ended state 505.

[0047] Also while in the completed state 503, the implementation computing system 201 may transmit a compensate message to the implementation computing system 202. This informs the implementation computing system 202 that the work being done should be

undone to the extent reasonable. Accordingly, the transmission of the compensate message from the implementation computing system 201 to the implementation computing system 202 results in a transition 513 from the completed state 503 to the compensating state 506

[0048] While in the compensating state 506, the implementation computing system 201 may receive a compensated message from the implementation computing system 202. This informs the implementation computing system 202 that the compensation action was successful. Accordingly, the transmission of the compensated message from the implementation computing system 202 to the implementation computing system 201 results in a transition 525 from the compensating state 506 to the ended state 505.

[0049] Alternatively, while in the compensating state 506, the implementation computing system 201 may receive a faulted message from the implementation computing system 202. This informs the implementation computing system 201 that the implementation computing system 202 has failed from the compensating state 506. Accordingly, the transmission of the faulted message from the implementation computing system 202 to the implementation computing system 201 results in a transition 527 from the compensating state 506 to a faulted state 507.

[0050] While in the faulted state 507, the implementation computing system 201 may transmit a forget message to the implementation computing system 202. This informs the implementation computing system 202 that the implementation computing system 201 is aware of the fault. Accordingly, the transmission of the forget message from the implementation computing system 201 to the implementation computing system 202 results in a transition 514 from the faulted state 507 to the ended state 505.

[0051] Figure 6 illustrates a flowchart of a method 600 for the code generation computing system 301 to automatically generate code that tests capabilities of the test

22

computing system 401 to use a message exchange pattern application to engage in message transactions following a message exchange pattern. The code permits the simulation without requiring that the test computing system 401 engage in the message transaction with other organizations, or even in some cases without requiring communication with other computing systems. The method 600 will be described with frequent reference to the specific example message exchange pattern definition of Figure 5.

[0052] The code generation computing system accesses a message exchange pattern definition (act 601) such as the message exchange pattern definition described above. The code generation computing system then performs a functional, result-oriented step for automatically generating message exchange pattern simulation code using the message exchange pattern definition (step 610). The step 610 may include any corresponding specific acts that accomplish this result. However, in the illustrated embodiment, the step 610 includes corresponding acts 611 and 612.

[0053] Specifically, for each state in which the message exchange pattern definition allows a valid transmission message to be transmitted by the test computing system, the code generation computing system automatically generates code that at least simulates transmission of the valid transmission message (act 611). For example, consider the active state 501 of Figure 5. The active state 501 permits one valid message to be transmitted, a Cancel message. There is indicated to be a 3% change that the cancel message will be transmitted. This percentage may be specified by the message exchange pattern definition and/or may be specified by the user or the code generation computing system. The percentage associated with this transmission may dynamically change over time during the testing process. To simulate the conditional transmission of the Cancel message, the code

22

generation computing system may generate code that follows the following pseudo-code document with line number added for ease of later explanation:

```

1.    main
2.    {
3.        Do State501( )
4.        {
5.            Flip a Coin
6.            if Coin is in 3% range
7.                Send(Cancel)
8.                Go to DoState502
9.            end if
10.        }
11.    .
12.    .
13.    .
14.    }

```

[0054] In this pseudo-code, a main program is created with its title at line 1, its opening bracket at line 2, and its closing bracket at line 14. Then, a subroutine is generated called DoState501. Line 3 identifies the subroutine which is bracketed using an open bracket at line 4 and a closing bracket at line 10. Code is then generated to generate a pseudo-random value (see line 5). Conditional code is then generated so that if appropriate given that random value (see line 6), the transmission of the cancel message is at least simulated (see

line 7), and code is generated to advance execution to the next state (see line 8) to thereby transition to the next appropriate state.

[0055] Returning back to Figure 6, for each state in which the message exchange pattern definition allows a valid receipt message to be received, the code generation computing system generates code that simulates the receipt of the valid receipt message, and that transitions to other code that represents the state to transition. For example, the active state 501 permits the receipt of three valid messages; an Exited message having a 10% chance of occurrence, a Completed message having a 75% chance of occurrence, and a Faulted message having a 2% change of occurrence. This is simulated in the example pseudo-code by adding additional conditional statements in the same DoState501 subroutine as follows:

```

1.  main
2.  {
3.      Do State501( )
4.      {
5.          Flip a Coin
6.          if Coin is in the appropriate 3% range
7.              Send(Cancel)
8.              Go to DoState502
9.          end if
10.         if Coin is in the appropriate 10% range
11.             Receive(Exited)
12.             Go to DoState505
13.         end if

```

```

14.         if Coin is in the appropriate 75% range
15.             Receive(Completed)
16.             Go to DoState503
17.         end if
18.         if Coin is in the appropriate 2% range
19.             Receive(Faulted)
20.             Go to DoState507
21.         end if
22.     }
23.     .
24.     .
25.     .
26. }

```

[0056] Note the addition of the conditionally executed pseudo-code for simulating receipt of the Exited message with appropriate state transition to state 505 (see lines 10-13), simulating receipt of the Completed message with appropriate state transition to state 503 (see lines 14-17), and simulating receipt of the Faulted message with appropriate state transition to state 507 (see lines 18-21).

[0057] In addition, the transmission or receipt of invalid messages may be simulated by assigning a positive probability of occurrence to the transmission or receipt of certain invalid messages. These invalid messages may not be specified by the message exchange pattern definition because they do not follow the message exchange pattern. Instead, they may be entered by a user or by the code generation computing system. For instance, while in the

active state 501, an invalid message may be transmitted as indicated by arrow 541 (having a 7% chance of occurrence). Also, an invalid message may be received as indicated by arrow 542 (having a 3% chance of occurrence). Simulating the receipt and transmission of invalid message allows the simulation to test the very real possibility of having to deal with non-conformity to the message exchange pattern. Similar code may be automatically generated for these invalid messages as well. For example, the following pseudo-code extends the previous example to generate code that conditionally simulates the transmission (see lines 25-27) and receipt (see lines 22-24) of invalid messages.

```

1.  main
2.  {
3.      Do State501( )
4.      {
5.          Flip a Coin
6.          if Coin is in the appropriate 3% range
7.              Send(Cancel)
8.              Go to DoState502
9.          end if
10.         if Coin is in the appropriate 10% range
11.             Receive(Exited)
12.             Go to DoState505
13.         end if
14.         if Coin is in the appropriate 75% range
15.             Receive(Completed)

```

12

```

16.          Go to DoState503
17.      end if
18.      if Coin is in the appropriate 2% range
19.          Receive(Faulted)
20.          Go to DoState507
21.      end if
22.      if Coin is in the appropriate 3% range
23.          Receive(Invalid Message)
24.      end if
25.      if Coin is in the appropriate 7% range
26.          Send(Invalid Message)
27.      end if
28.  }
29.  .
30.  .
31.  .
32.  }
```

[0058] This generation process may be also performed to generate subroutines for the other states as well (e.g., states 502 through 507), which may also have their own branching probabilities. The code generation process involves generating a primary program (e.g., the program "main"), generating a subroutine for each state, generating a random number for each state in which multiple events may occur, and then conditionally performing the event based on the random number. As this is a highly procedural process, involving little in the

22

way of judgment calls, the code generation process may be highly automated. Any user intervention, if any, would be trivial. For example, the user may need to estimate probabilities of occurrence for each valid message type, and indicate whether or not invalid message receipt or transmission may occur (along with associated probabilities). However, the user would not actually need to draft the code itself. Accordingly, code is more easily generated.

[0059] Furthermore, the "send" and "receive" functions may be authored to merely simulate the transmission and receipt of messages as they would occur during actual deployment. Accordingly, there is no need to rely on the cooperation of the other organization that will ultimately be involved with the message exchange pattern in order to test the message exchange pattern application. In one embodiment, the transmission and reception itself may be simulated such that the simulation results in no actual network traffic, thereby preserving network resources.

[0060] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes, which come within the meaning and range of equivalency of the claims, are to be embraced within their scope.

[0061] What is claimed and desired secured by United States Letters Patent is:

1. In a code generation computing system that includes one or more processors capable of executing computer-executable instructions in system memory, a method for the code generation computing system to automatically generate code that tests capabilities of a test computing system to use a message exchange pattern application to engage in message transactions following a message exchange pattern, the method comprising the following:

an act of accessing a message exchange pattern definition that defines a plurality of states, the message exchange pattern definition further having for each of the plurality of states, an indication of one or more valid messages that conform to the message exchange pattern for that state, an indication of which computing system may transmit each valid message given the state, and a state transition indication for at least some of the valid messages identifying one of the plurality of states to transition to should the valid message occur;

for each state in which the message exchange pattern definition allows a valid transmission message to be transmitted by the test computing system, an act of generating code that at least simulates transmission of the valid transmission message, and that transitions to other code that represents the state to transition to should the valid transmission message be transmitted; and

for each state in which the message exchange pattern definition allows a valid receipt message to be received, an act of generating code that simulates the receipt of the valid receipt message, and that transitions to other code that represents the state to transition to should the valid receipt message be received.

2. A method in accordance with Claim 1, wherein the message exchange pattern definition indicates that when in a particular state, any one of a plurality of valid transmission messages may be transmitted.

3. A method in accordance with Claim 2, wherein the message exchange pattern definition indicates for each of the plurality of valid transmission messages for the particular state, a percentage chance that each of the plurality of valid transmission messages will be transmitted given the particular state, wherein the method further comprises automatically generating code for at least simulating each of the plurality of valid transmission messages and performing appropriate state transitions given the transmission.

4. A method in accordance with Claim 3, further comprising the following:
an act of generating code that generates a pseudo-random value and selects one of the plurality of valid transmission messages for transmission based on the pseudo-random value and on the percentage chance.

5. A method in accordance with Claim 1, further comprising the following for at least one state:

an act of generating code that at least simulates transmission of an invalid transmission message.

6. A method in accordance with Claim 5, wherein there is also indicated a percentage chance that the invalid transmission messages will be transmitted given the

particular state, wherein the method further comprises an act of generating code that generates a pseudo-random value and selects one of the invalid transmission message for transmission based on the pseudo-random value and on the percentage chance.

7. A method in accordance with Claim 1, wherein the message exchange pattern definition indicates that when in a particular state, any one of a plurality of valid receipt messages may be received.

8. A method in accordance with Claim 7, wherein the message exchange pattern definition indicates for each of the plurality of valid receipt messages for the particular state, a percentage chance that each of the plurality of valid receipt messages will be received given the particular state, wherein the method further comprises automatically generating code for simulating each of the plurality of valid receipt messages and performing appropriate state transitions given the transmission.

9. A method in accordance with Claim 8, further comprising the following:
an act of generating code that generates a pseudo-random value and selects one of the plurality of valid receipt messages for simulated receipt based on the pseudo-random value and on the percentage chance.

10. A method in accordance with Claim 1, further comprising the following for at least one state:

an act of generating code that simulates receipt of an invalid transmission message.

32

11. A method in accordance with Claim 10, wherein there is also indicated a percentage chance that the invalid receipt message will be received given the particular state, wherein the method further comprises an act of generating code that generates a pseudo-random value and selects one of the invalid received message for simulated receipt based on the pseudo-random value and on the percentage chance.

12. A method in accordance with Claim 1, wherein the message exchange pattern definition is defined using the Web Services Description Language (WSDL) standard.

13. A method in accordance with Claim 1, wherein the message exchange pattern definition further defines timing policies to be imposed when in a particular state.

32

14. A computer program product for use in a code generation computing system that includes one or more processors capable of executing computer-executable instructions in system memory, the computer program product for performing a method for the code generation computing system to automatically generate code that tests capabilities of a test computing system to use a message exchange pattern application to engage in message transactions following a message exchange pattern, the computer program product comprising one of more computer-readable media that, when executed by one or more processors of the code generation computing system, causes the code generation computing system to perform the method, the method comprising the following:

an act of accessing a message exchange pattern definition that defines a plurality of states, the message exchange pattern definition further having for each of the plurality of states, an indication of one or more valid messages that conform to the message exchange pattern for that state, an indication of which computing system may transmit each valid message given the state, and a state transition indication for at least some of the valid messages identifying one of the plurality of states to transition to should the valid message occur;

for each state in which the message exchange pattern definition allows a valid transmission message to be transmitted by the test computing system, an act of generating code that at least simulates transmission of the valid transmission message, and that transitions to other code that represents the state to transition to should the valid transmission message be transmitted; and

for each state in which the message exchange pattern definition allows a valid receipt message to be received, an act of generating code that simulates the receipt of the valid

receipt message, and that transitions to other code that represents the state to transition to should the valid receipt message be received.

15. A computer program product in accordance with Claim 14, wherein the message exchange pattern definition indicates that when in a particular state, any one of a plurality of valid transmission messages may be transmitted.

16. A computer program product in accordance with Claim 15, wherein the message exchange pattern definition indicates for each of the plurality of valid transmission messages for the particular state, a percentage chance that each of the plurality of valid transmission messages will be transmitted given the particular state, wherein the method further comprises automatically generating code for at least simulating each of the plurality of valid transmission messages and performing appropriate state transitions given the transmission.

17. A computer program product in accordance with Claim 16, further comprising the following:

an act of generating code that generates a pseudo-random value and selects one of the plurality of valid transmission messages for transmission based on the pseudo-random value and on the percentage chance.

18. A computer program product in accordance with Claim 14, further comprising the following for at least one state:

an act of generating code that at least simulates transmission of an invalid transmission message.

19. A computer program product in accordance with Claim 18, wherein there is also indicated a percentage chance that the invalid transmission messages will be transmitted given the particular state, wherein the method further comprises an act of generating code that generates a pseudo-random value and selects one of the invalid transmission message for transmission based on the pseudo-random value and on the percentage chance.

20. A computer program product in accordance with Claim 14, wherein the message exchange pattern definition indicates that when in a particular state, any one of a plurality of valid receipt messages may be received.

21. A computer program product in accordance with Claim 20, wherein the message exchange pattern definition indicates for each of the plurality of valid receipt messages for the particular state, a percentage chance that each of the plurality of valid receipt messages will be received given the particular state, wherein the method further comprises automatically generating code for simulating each of the plurality of valid receipt messages and performing appropriate state transitions given the transmission.

22. A computer program product in accordance with Claim 21, further comprising the following:

36

an act of generating code that generates a pseudo-random value and selects one of the plurality of valid receipt messages for simulated receipt based on the pseudo-random value and on the percentage chance.

23. A computer program product in accordance with Claim 14, further comprising the following for at least one state:

an act of generating code that simulates receipt of an invalid transmission message.

24. A computer program product in accordance with Claim 23, wherein there is also indicated a percentage chance that the invalid receipt message will be received given the particular state, wherein the method further comprises an act of generating code that generates a pseudo-random value and selects one of the invalid received message for simulated receipt based on the pseudo-random value and on the percentage chance.

25. A computer program product in accordance with Claim 14, wherein the message exchange pattern definition is defined using the Web Services Description Language (WSDL) standard.

26. A computer program product in accordance with Claim 14, wherein the message exchange pattern definition further defines timing policies to be imposed when in a particular state.

27. A computer program product in accordance with Claim 14, wherein the one or more computer-readable media are physical media.

28. A computer program product in accordance with Claim 27, wherein the one or more computer-readable media includes system memory.

29. A computer program product in accordance with Claim 27, wherein the one or more computer-readable media includes persistent memory.

30. A computer program product in accordance with Claim 29, wherein the persistent memory is a magnetic disk.

31. In a code generation computing system that includes one or more processors capable of executing computer-executable instructions in system memory, a method for the code generation computing system to automatically generate code that tests capabilities of a test computing system to use a message exchange pattern application to engage in message transactions following a message exchange pattern, the method comprising the following:

an act of accessing a message exchange pattern definition that defines a plurality of states, the message exchange pattern definition further having for each of the plurality of states, an indication of one or more valid messages that conform to the message exchange pattern for that state, an indication of which computing system may transmit each valid message given the state, and a state transition indication for at least some of the valid messages identifying one of the plurality of states to transition to should the valid message occur; and

a step for automatically generating message exchange pattern simulation code using the message exchange pattern definition.

32. A method in accordance with Claim 31, wherein the step for automatically generating message exchange pattern simulation code using the message exchange pattern definition comprises the following:

for each state in which the message exchange pattern definition allows a valid transmission message to be transmitted by the test computing system, an act of generating code that at least simulates transmission of the valid transmission message, and that transitions to other code that represents the state to transition to should the valid transmission message be transmitted; and

for each state in which the message exchange pattern definition allows a valid receipt message to be received, an act of generating code that simulates the receipt of the valid receipt message, and that transitions to other code that represents the state to transition to should the valid receipt message be received.

ABSTRACT OF THE DISCLOSURE

Mechanisms for automatically generating code that tests capabilities of a test computing system to simulate a message exchange pattern. The code generation computing system uses a message exchange pattern definition to generate the simulation code. For each state in which the message exchange pattern definition allows valid messages to be transmitted, code is generated for that state that at least simulates the transmission of a valid transmission message. For each state in which the message exchange pattern definition allows valid messages to be received, code is generated for that state that simulates the receipt of a valid receipt message. If the transmission or receipt of the message causes a state transition to occur, code is generated that causes the appropriate state transition in the message exchange pattern.

| v Eliminado: 1

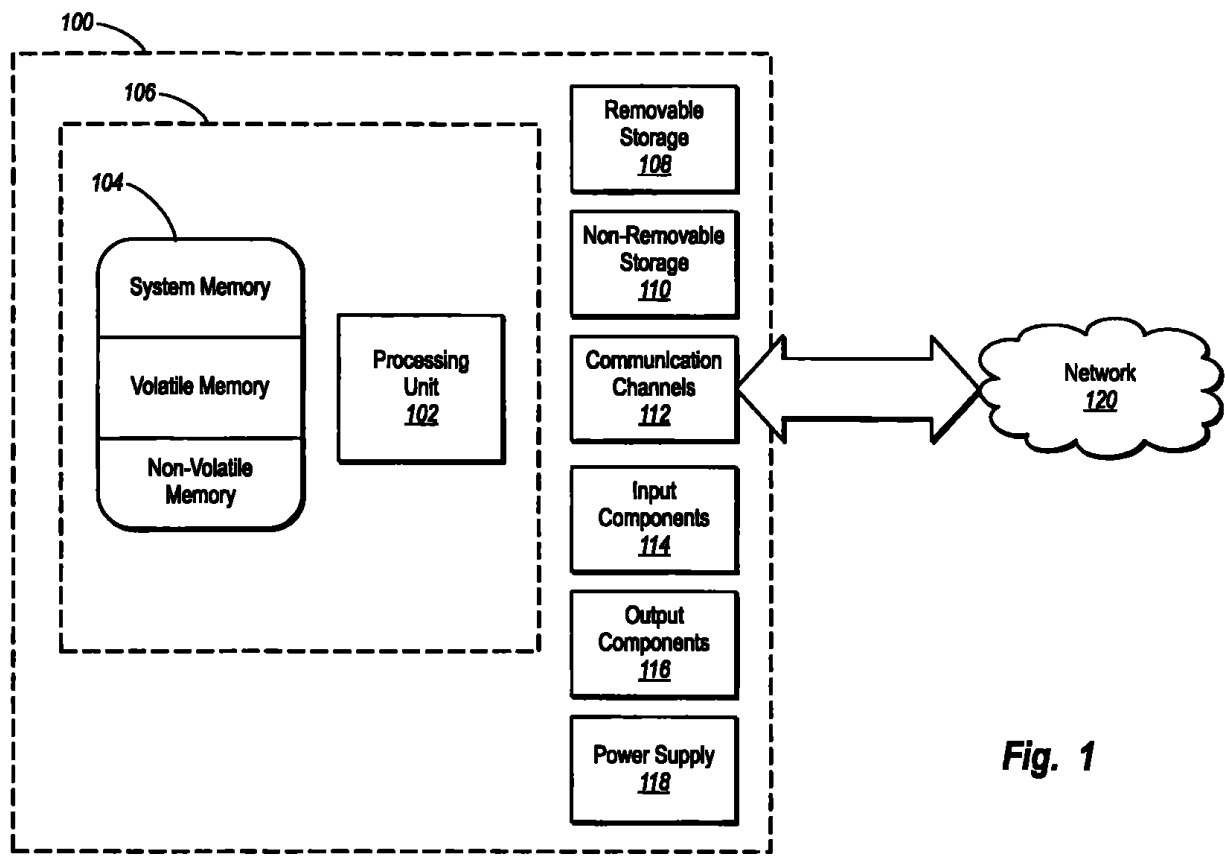


Fig. 1

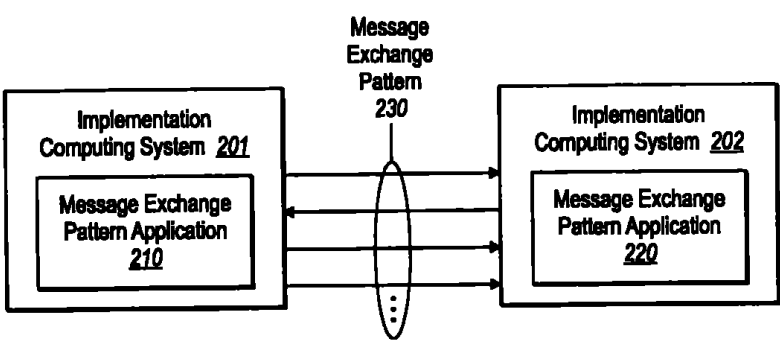


Fig. 2

2/4

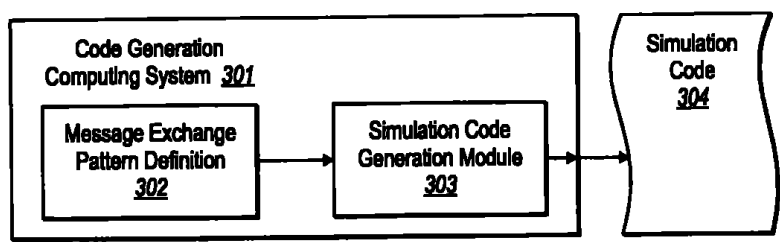


Fig. 3

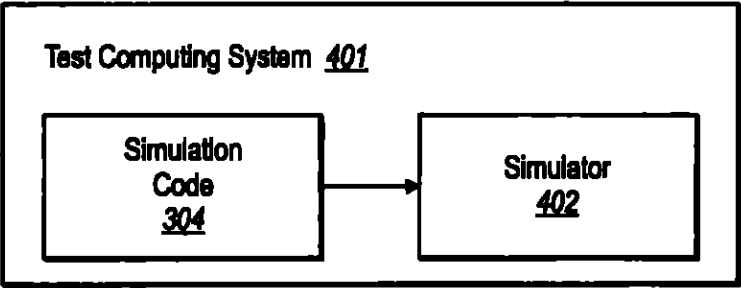


Fig. 4

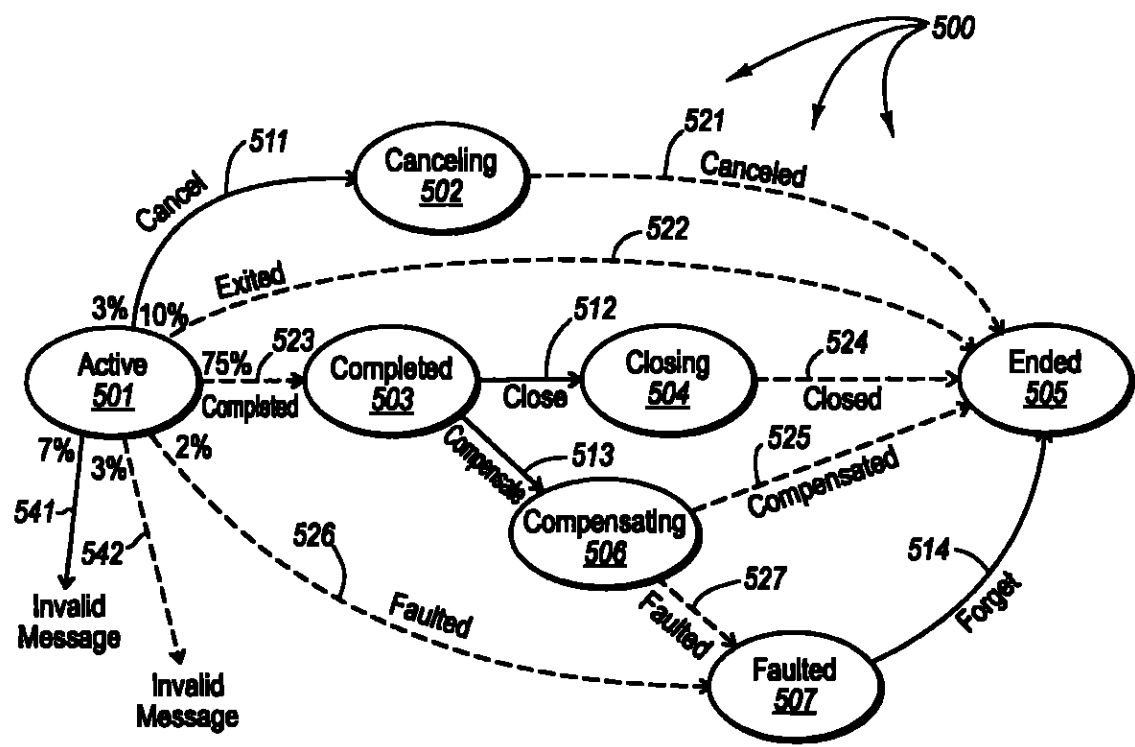


Fig. 5

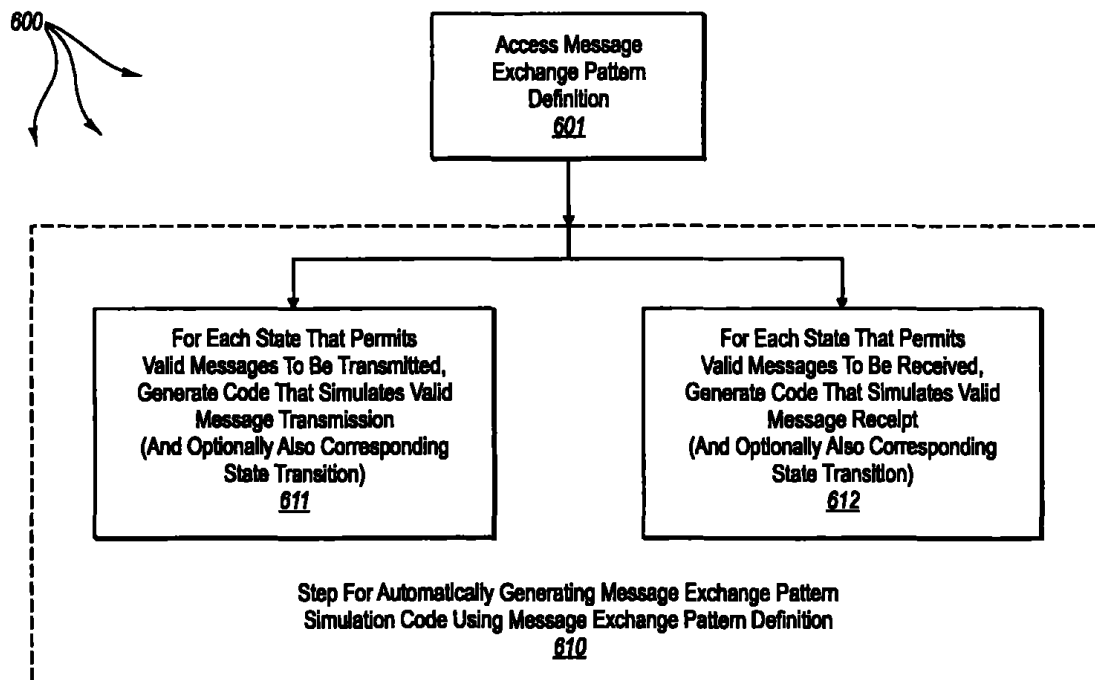


Fig. 6

GENERACIÓN AUTOMATIZADA DE CÓDIGO DE SIMULACIÓN DE PATRONES DE INTERCAMBIO DE MENSAJES

1. Campo de la Invención

La presente invención se relaciona con tecnologías en red; y más específicamente, a la generación automatizada de un código que simula un patrón de intercambio de mensajes.

2. Antecedentes y Estado de la Técnica

La tecnología del cómputo ha transformado la manera en que trabajamos y jugamos. Los sistemas de cómputo toman ahora una amplia variedad de formas incluyendo computadores de mesa, computadores portátiles, PC de tableta, Asistentes Digitales Personales (PDA), y similares. Aún los dispositivos domésticos (tales como refrigeradores, hornos, máquinas de coser, sistemas de seguridad, y similares) tienen niveles variables de capacidad de procesamiento y por lo tanto pueden ser considerados sistemas de cómputo.

Mucha de la funcionalidad provista por sistemas de cómputo descansa en que el sistema de cómputo esté en red con (y ser capaz de comunicar mensajes con) otros sistemas de cómputo. Para que los dos sistemas de cómputo en comunicación puedan llevar a cabo una tarea particular, una transacción de mensajes que involucra un patrón de intercambio de mensajes puede requerirse. Cada sistema de cómputo involucrado en la transacción de mensajes tiene una aplicación (llamada de aquí en adelante "una aplicación de patrón de intercambio de mensaje") que ayuda a procesar los mensajes recibidos como parte de la transacción que sigue el patrón de intercambio de mensajes, y transmitir otros mensajes que se conforman con el patrón de intercambio de mensajes.

Para poder ensayar una aplicación de patrón de intercambio de mensajes, la aplicación de patrón de intercambio de mensajes debe estar instalada en diferentes sistemas de cómputo y entonces verificad. Sin embargo, en muchos casos, la aplicación de patrón de intercambio de mensajes en un sistema de cómputo puede diferir significativamente de la aplicación de patrón de intercambio de mensajes en otro sistema de cómputo, aún cuando las dos aplicaciones de patrón de intercambio de

mensajes sean capaces de interactuar una con la otra usando el patrón de intercambio de mensajes. Más aún, el propietario de otra aplicación de patrón de intercambio de mensajes puede no querer o ser capaz de evaluar cooperativamente la interacción de las dos aplicaciones de patrón de intercambio de mensajes. De acuerdo con lo anterior, no siempre es factible evaluar la aplicación de patrón de intercambio de mensajes en un ambiente de red real en el cual será implementada y desplegada.

Lo que sería ventajoso es un mecanismo para ensayar o evaluar aplicaciones de patrón de intercambio de mensajes sin tener que evaluar la aplicación en el ambiente de red real en el cual esta será finalmente empleada. Sería además muy ventajoso si la evaluación puede ser llevada a cabo sin requerir la autorización de software de evaluación adicional cada vez que una aplicación de patrón de intercambio de mensajes va a ser evaluada.

BREVE RESUMEN DE LA INVENCION

Los problemas anteriores con el estado de la técnica se solucionan por los principios de la presente invención, los cuales están dirigidos hacia mecanismos para generar

automáticamente el código que evalúe las capacidades de un sistema de cómputo de ensayo para simular un patrón de intercambio de mensajes.

El sistema de cómputo de generación de códigos usa una definición de patrón de intercambio de mensajes para generar el código de simulación. La definición de patrón de intercambio de mensajes incluye lo siguiente para cada uno de los estados de transición en el patrón de intercambio de mensajes: 1) una indicación de cual sistema de cómputo puede transmitir (y/o recibir) cada mensaje válido dado el estado, y 2) una descripción de transición de estado que resulta de transmitir o recibir un mensaje válido.

Para cada estado en el cual una definición de patrón de intercambio de mensajes permite que se trasmitan mensajes válidos, el código es generado para ese estado de modo que por lo menos simule la transmisión de un mensaje de transmisión válido. Para cada estado en el cual la definición de patrón de intercambio de mensajes permite que se reciban mensajes válidos, el código es generado para ese estado de modo por lo menos simule la recepción de un mensaje recibido válido. Si la transmisión o

recepción del mensaje causa que ocurra un estado de transición, un código es generado que cause la transición de estado apropiada en el patrón de intercambio de mensajes. El sistema de cómputo de generación de códigos puede también generar el código que simule la transmisión o recepción de mensajes inválidos dado un estado particular también. Se pueden asignar probabilidades a cierto(s) mensaje(s) (sean válido o inválidos) que está siendo transmitido o recibido. El patrón de intercambio de mensajes puede ser simulado un número de veces para simular muchos o todos los trayectos de transición de estado posibles por el patrón de intercambio de mensajes. Esto permite que la simulación cubra aún potencialmente todos los casos posibles que se encontrarán en el despliegue real.

De acuerdo con lo anterior, el código de simulación es generado automáticamente usando la definición del patrón de intercambio de mensajes, haciendo por lo tanto más fácil generar el código de simulación. Más aún, la simulación puede ser interna en un único sistema de cómputo ya que la transmisión y recepción de mensajes puede ser simulada sin que exista comunicación con otro sistema de cómputo. De acuerdo con lo anterior, no hay

necesidad del uso de recursos de red. Alternativamente, aún si la simulación no es interna a un único sistema de cómputo, la simulación permite la comunicación al interior de la red de una única organización. De acuerdo con lo anterior, no hay necesidad de recostarse en la cooperación de otra organización para poder evaluar la aplicación de patrón de intercambio de mensajes.

Rasgos y ventajas adicionales de la invención se establecerán en la descripción que sigue, y en parte serán obvias a partir de la descripción, o pueden ser aprendidas por la práctica de la invención. Los rasgos y ventajas de la invención pueden lograrse y obtenerse por medio de instrumentos y combinaciones destacadas particularmente en las reivindicaciones adjuntas. Estos y otros rasgos de la presente invención serán más y más evidentes de la siguiente descripción y reivindicaciones adjuntas, o pueden ser aprendidos por la práctica de la invención según se establece de aquí en adelante.

BREVE DESCRIPCIÓN DE LOS DIBUJOS

Para poder describir la manera en la cual las ventajas y rasgos mencionados arriba y otros de la invención pueden

ser obtenidos, una descripción más particular de la invención brevemente descrita antes se dará con referencia a modalidades específicas de la misma las cuales se ilustran en los dibujos adjuntos. Se entiende que estos dibujos ilustran solamente modalidades típicas de la invención y por lo tanto no deben ser considerados limitativos de su alcance, la invención será descrita y explicada con especificidad y detalle adicional usando los dibujos adjuntos en los cuales:

La figura 1 ilustra un sistema de cómputo adecuado que puede implementar rasgos de la presente invención;

La figura 2 ilustra un ambiente de red en el cual dos sistemas de cómputo implementan un patrón de intercambio de mensajes, cada uno usando una aplicación de patrón de intercambio de mensajes;

La figura 3 ilustra un sistema de cómputo de generación de códigos que genera automáticamente el código que simula internamente un patrón de intercambio de mensajes en un sistema de cómputo de ensayo;

La figura 4 ilustra un sistema de cómputo de prueba o ensayo que simula internamente el patrón de intercambio de mensajes usando el código de simulación generado por el sistema de cómputo de generación de códigos;

La figura 5 ilustra un árbol de transición de estados que puede ser usado para rastrear el progreso en un patrón de intercambio de mensajes arbitrario; y

La figura 6 ilustra un diagrama de flujo de un método para que el sistema de cómputo de generación de códigos genere automáticamente el código de simulación que simula internamente un patrón de intercambio de mensajes.

DESCRIPCIÓN DETALLADA DE LAS MODALIDADES PREFERIDAS

Los principios de la presente invención se relacionan con mecanismos para generar automáticamente códigos que prueban las capacidades de un sistema de cómputo de ensayo para simular un patrón de intercambio de mensaje. El sistema de cómputo de generación de códigos utiliza una definición de patrón de intercambio de mensajes como una base para generar automáticamente el código de simulación. Para cada estado en el cual la definición del patrón de intercambio de mensajes permite que los mensajes válidos sean transmitidos, el código es generado para ese estado que por lo menos simula la transmisión de un mensaje de transmisión válido. Para cada estado en el cual la definición del patrón de intercambio de mensajes permite que mensajes válidos sean recibidos, el código es generado para ese estado que simula la recepción de un mensaje de recepción válido. Si la transmisión o recepción del mensaje causa una transición de estado, el código es generado de modo que cause la transición de estado apropiada en el patrón de intercambio de mensajes.

Volviendo a los dibujos, en donde los numerales de referencias similares se refieren a elementos similares,

la invención es ilustrada como implementada en un ambiente de cómputo adecuado. La siguiente descripción está basada en las modalidades de la invención y no debe ser tomada como limitante de la invención con respecto a las modalidades alternativas que no son explícitamente descritas aquí.

En la descripción que sigue, la invención es descrita con referencia a actos y representaciones simbólicas que son llevadas a cabo por uno o más computadores a menos que se indique lo contrario. Como tal, debe entenderse que tales actos de operaciones, que son referidos como siendo ejecutadas por un computador, incluyen la manipulación por la unidad de procesamiento del computador de señales eléctricas que representan datos en una forma estructurada. Esta manipulación transforma los datos o los mantiene en localizaciones en el sistema de memoria del computador, el cual reconfigura o de otra manera altera la operación del computador de una manera bien entendida por aquellos versados en la técnica. Las estructuras de datos en donde los datos son mantenidos son localizaciones físicas de la memoria que tienen propiedades particulares definidas por el formato de los datos. Sin embargo, aunque la invención está siendo

descrita en el contexto anterior, no significa que es limitada ya que aquellos basados en la técnica podrán apreciar que diversos varios de los actos y operaciones descritos de aquí en adelante pueden también ser incrementados en hardware. La figura 1 muestra un diagrama esquemático de una arquitectura de computador ejemplo utilizable para estos dispositivos.

Para propósitos descriptivos la arquitectura mostrada es solamente un ejemplo de un ambiente adecuado y no tiene el propósito de sugerir ninguna limitación en cuanto al alcance del uso o funcionalidad de la invención. Tampoco deberían los sistemas de cómputo ser interpretado como que tiene cualquier dependencia o requerimiento con relación a cualquiera o a la combinación de componentes ilustrados en la figura 1.

La invención es operacional con numerosos otros ambientes o configuraciones de comunicaciones o de cómputo de propósito especial. Ejemplos de sistemas de cálculos bien conocidos, ambientes, y configuraciones adecuadas para uso con la invención incluyen, pero no se limitan a, teléfonos móviles, computadores de bolsillo, computadores personales, servidores, sistemas multiprocesadores,

sistemas con base en microprocesador, mini computadores, computadores de tamaño medío, y ambientes de cómputo distribuidos que incluyen cualquiera de los anteriores sistemas o dispositivos.

En su configuración más básica, un sistema de cómputo científicamente incluye por lo menos una unidad de procesamiento 102 y una memoria 104. La memoria 104 puede ser volátil (tal como RAM), no volátil (tal como ROM, una memoria FLASH, etc.), o alguna combinación de las dos. Esta configuración más básica es ilustrada en la figura uno por la línea punteada 106.

Los dispositivos de medio de almacenamiento pueden tener características adicionales y funcionalidades adicionales. Por ejemplo, ellos pueden incluir almacenamiento adicional (removible y no removible) incluyendo, pero sin limitarse a, tarjetas PCMCIA, discos magnéticos y ópticos y cintas magnéticas. Tal almacenamiento adicional es ilustrado en la figura 1 por el almacenamiento removible 108 y el almacenamiento no removible 110. El medio de almacenamiento del computador incluye medios removibles y medios no removibles,

volátiles y no volátiles implementados en cualquier método o tecnología para el almacenamiento de información tal como instrucciones legibles por computadora, estructuras de datos, módulos de programas u otros datos. La memoria 104, el almacenamiento removible 108, y el almacenamiento no removible 110 son todos ejemplos de almacenamiento por computador. En los medios de almacenamiento por computador incluyen, pero no se limitan a RAM, ROM, EEPROM, memoria flash u otra tecnología de memoria, CD-ROM discos versátiles digitales, otros almacenamientos ópticos, cassettes magnéticos, cintas magnéticas, almacenamiento por disco magnético, u otros dispositivos de almacenamiento magnético, y cualquier otro medio que pueda ser usado para almacenar la información deseada y que pueda ser accesado por el sistema de cómputo.

Como se usa aquí, el término "modulo" o "componentes" puede referirse a objetos de software o rutinas que se ejecutan sobre el sistema de cómputo. Los diferentes componentes, módulos, motores, y servicios descritos aquí pueden ser implementados como objetos o procesos que se ejecutan en un sistema de cómputo (como hilos separados). Aunque el sistema y los métodos descritos aquí son

implementados preferiblemente en software, las implementaciones en software y hardware también son posibles y contempladas.

El sistema de cómputo 100 puede también contener canales de comunicación 112 que le permiten al anfitrión comunicarse con otros sistemas y dispositivos sobre la red 120. Los canales de comunicación 112 son ejemplos de medio de comunicación. Los medios de comunicación típicamente involucran instrucciones legibles por computadora, estructuras de datos, módulos de programas u otros datos en una señal de datos modulada tal como una onda portadora u otro mecanismo de transporte e incluye cualesquier medio de suministro de información. Por vía de ejemplo, y sin limitación, los medios de comunicación incluyen medios alambrados, tales como redes alambradas y conexiones alambradas directamente, y medios inalámbricos tales como medios acústicos, de radio, infrarrojo u otros medios inalámbricos. El término medio legible por computadora como se usa aquí incluye tanto medios de almacenamiento como medios de comunicaciones.

El sistema de cómputo 100 puede también tener componentes de entrada 114 tales como teclado, ratón, lápiz, un

componente de entrada por voz, un componente de entrada por toque, etc. Los componentes de salida 116 incluyen visualizadores de pantalla, parlantes, impresoras, etc., y convierten a los módulos (frecuentemente llamados "adaptadores") para manejarlos. El sistema de cómputo 100 tiene una fuente de suministro de potencia 118. Todos estos componentes son bien conocidos en la técnica y no requieren ninguna discusión larga aquí.

La figura dos ilustra un ambiente de red 200 en el cual dos sistemas de cómputo (referidos como "sistemas de cómputo de implementación" en la figura dos) se comunican para llevar a cabo una tarea particular. En particular, el sistema de cómputo de implementación 201 se comunica con el sistema de cómputo de implementación 202. Para poder llevar a cabo la tarea particular, los sistemas de cómputo 201 y 202 intercambian mensajes en un patrón particular de intercambio 230. El patrón particular de intercambio de mensajes define cual sistema de cómputo va a enviar que mensaje en un punto dado en el intercambio de mensajes. El patrón de intercambio de mensajes depende de la tarea que se va llevar a cabo y los protocolos usados para intercambiar mensajes. Los principios de la

presente invención no están limitados para el patrón de intercambio de mensajes.

Cada sistema de cómputo de implementación tiene una aplicación de patrón de intercambio de mensajes que engancha en el patrón intercambio de mensajes. Por ejemplo, el sistema de cómputo de implementación 201 ejecuta la aplicación de patrón de intercambio de mensaje 210 mientras que el sistema de cómputo de implementación 202 ejecuta la aplicación del patrón de intercambio de mensaje 220. Los sistemas de cómputo de implementación 201 y 202 pueden estar estructurados como se describió anteriormente para el sistema de cómputo aunque esto no es requerido. En esta descripción y en la reivindicación, un "sistema de cómputo" es definido como cualquier dispositivo o sistema que tiene la capacidad de procesar mensajes electrónicos.

Los mensajes dentro del patrón de intercambio de mensajes no requieren ser todos generados y consumidos por los mismos componentes en el sistema de cómputo. Por ejemplo algunos mensajes pueden ser mensajes del nivel de protocolo que permitan la coordinación apropiada en el nivel protocolo entre los dos sistemas de cómputo. Otros

mensajes pueden ser mensajes de nivel de aplicación para permitir el intercambio de datos requerido para llevar a cabo la tarea deseada. En otra modalidad los mensajes de nivel de protocolo en el patrón de intercambio de mensaje están gobernados por la coordinación de servicios de red (ws-coordinación) y las especificaciones de transacción ws, que son conocidas para aquellos versados en la técnica.

La figura 2 ilustra un el ambiente en el cual las aplicaciones de patrón de intercambio de mensajes operaran finalmente. Sin embargo, es deseable verificar o evaluar la aplicación de patrón de intercambio de mensajes para asegurar una operación apropiada antes de andar la aplicaron. Frecuentemente toma tiempo considerable ensayar la aplicaciones ya que frecuentemente los nuevos códigos requieren ser diseñados para poder llevar a cabo la simulación. más aún el ensayo o evaluación de las aplicaciones de patrón de Intercambio de mensajes es particularmente difícil ya que las operaciones son por naturaleza distribuidas, y frecuentemente distribuidas atrases de límites de confianza. Por ejemplo, aquellos desarrolladores de la aplicación de patrón de intercambio de mensaje 210 pueden

no tener acceso a la aplicación de intercambio de mensaje 220 que es la que corre el otro sistema de cómputo. Más aún la cooperación del propietario de la aplicación del patrón de intercambio de mensaje 220 puede ser difícil de adquirir para propósitos de ensayo.

De acuerdo con lo anterior, los principios de la presente invención suministran un mecanismo en donde el código es generado automáticamente lo cual permite a un sistema de cómputo de evaluación simular el conducto de un patrón de intercambio de mensajes sin requerir la autorización humana extensiva del código, y sin requerir que la prueba ocurra con la copelación de otra organización o aún quizás en cualquier ambiente de red. La figura 3 ilustra un también en el cual tal generación de código automatizado puede ocurrir.

Haciendo referencia a la figura 3, un sistema de cómputo de generación de códigos 301 accesa una definición de patrón de intercambio de mensajes 302. Un módulo de generación de código de simulación 303 usa la definición del patrón de intercambio de mensajes 303 para generar el código de simulación 304. Refiriéndonos a la figura 4 este código de simulación puede entonces ser usado por un

sistema de cómputo de prueba 401 por un simulador 402 para ensayar la definición de intercambio de mensajes. Más con respecto a esta operación será descrito más adelante después de primero describir la definición del patrón de intercambio de mensajes 303 en mayor detalle.

En el patrón de intercambio de mensajes, hay un número de estados. La definición de patrón de intercambio de mensajes define estos estados. Para cada estado, habrá 0 o más tipos de mensajes válidos que puedan ser transmitidos y 0 o más tipos de mensajes válidos que puedan ser recibidos. Con la recepción o transmisión de un tipo de mensaje válido, el patrón de intercambio de mensajes puede hacer transición a un estado diferente. El patrón de intercambio de mensajes indica los tipos de mensajes válidos para cada estado, e indica que transiciones de estado deben ocurrir dado un cierto tipo de mensaje válido. La definición de patrón de intercambio de mensajes puede estar en cualquier formato. Sin embargo, para poder interpretar el patrón de intercambio de mensajes y para poder generar código automáticamente es ventajoso si la definición de intercambio de mensajes está en un formato que pueda ser más fácilmente

analizado. Un ejemplo de tal formato está definido por el estándar del lenguaje de descripción de servicios (WSDL).

La definición del intercambio de mensajes también puede incluir, por ejemplo, probabilidades de ocurrencia de que un cierto mensaje válido será recibido o transmitido dado un cierto estado. En la definición del patrón de intercambio de mensajes puede también imponer políticas de temporización que deberían ser aplicadas dado un cierto estado. Por ejemplo, políticas de temporización pueden indicar que una requisición previa debería ser cancelada si no hay respuesta recibida dentro de un cierto periodo de tiempo.

La definición del patrón de intercambio de mensajes puede ser mejor entendida reduciendo la definición de intercambio de mensajes a un árbol de transición de estados. Un árbol de transición de estados puede ser construido en la memoria durante el proceso de generación de códigos aunque esto no necesita ser el caso. El árbol de transición de estado del patrón de intercambio de mensajes incluye nodos para cada estado en el patrón de intercambio de mensajes. Para cada nodo, se ha suministrado un listado de tipos de mensajes válido y que

el sistema de cómputo puede enviar que tipos de mensajes. Para cada tipo de mensaje válido, hay también una identificación de que estado para transición y cuando se recibe (o transmisión como sea el caso) el mensaje especificado.

La estructura del árbol depende completamente del patrón de intercambio de mensajes lo que a su vez depende de la tarea que va a llevarse a cabo. Ilustrando todos los árboles de transición de estados posibles, sería imposible debido a la variedad sin fin de patrones de intercambio de mensajes. Listar muchos oscurecería los principios de la presente invención. Sin embargo, un patrón de intercambio de mensajes es descrito aquí para propósitos ilustrativos solamente. Aquellos versados en la técnica podrán apreciar (después de haber leído esta descripción) que los principios de la presente invención aplican a cualquier patrón de intercambio de mensaje.

Esa dicha, figura 5 ilustra un árbol de transición de estado de el patrón de intercambio de mensajes 500. El árbol de transición de estados 500 incluye un número de estados 501 a 507. La transmisión de mensajes de uno de los sistemas de cómputo resulta en transiciones de estado

representadas por las flechas sólidas. Por otro lado, la transmisión del mensaje desde otro de los sistemas de cómputo resulta en transiciones de estado representadas por las flechas punteadas. Por ahora ignore las flechas 541 y 542 lo mismo que las marcas de porcentaje, dichos elementos serán explicados más adelante con respecto a la figura 6. Haciendo referencia a la figura 5 un sistema de cómputo transmite mensajes que resultan en transiciones de estado 511 a 514, mientras que el otro sistema de cómputo transmite mensajes que resultan en transiciones de estado 521 a 527. Cada flecha está asociada con un tipo de mensaje particular o grupo de tipos de mensaje que son válidos para esa transición.

Si la coordinación WS está siendo usada, es sistema de cómputo de implementación 201 puede primero enviar un mensaje de contexto de coordinación al sistema de cómputo de implementación 202. El sistema de cómputo de implementación 202 entonces puede enviar un mensaje de registro al sistema de cómputo de implementación 201. El sistema de cómputo de implementación 201 entonces enviar un mensaje de recepción de registro al sistema de cómputo de implementación 202. En este estado el sistema de cómputo de implementación 201 toma el papel de

"coordinador" y el sistema de cómputo de implementación 201 toma el papel de "participante", y el intercambio de mensaje está en el estado activo 501. Las líneas sólidas en la figura 5 representan las transiciones de estado causadas por el coordinador (es decir el sistema de cómputo de implementación 201) al transmitir mensaje, mientras que las líneas punteadas en la figura 5 representan transiciones de estado causadas por el participante (es decir el sistema de cómputo de implementación 202) transmitiendo mensajes.

Del estado activo 501, el sistema de cómputo de implementación 201 puede transmitir un mensaje de cancelación al sistema de cómputo de incremento 202. Esto representa que el sistema de cómputo de incremento 201 ha escogido cancelar las operaciones previamente solicitadas y resulta en una transición del estado activo 501 al estado de cancelación 502 según se ha representado por la flecha sólida 511. Más generalmente, el sistema de cómputo que transmite reconoce las transiciones de estado tan pronto como el mensaje asociado con la transición es transmitido. El sistema de cómputo receptor reconoce en las transiciones de estado tan pronto como el mensaje asociado con la transición es recibido. De acuerdo con lo

anterior, puede haber algunas diferencias transcientes en el progreso rastreando de la carpa de transición de estados según se ha reconocido por el sistema de cómputo de implementación 201, y el progreso rastreado de la capa de transición de estados según se reconoce por el sistema de cómputo de implementación 202. En esta transición específica 511, el sistema de cómputo de implementación 201 reconoce esta transición 511 tan pronto como este transmite el mensaje de cancelación. El sistema de cómputo y de implementación 202 reconoce la transición 511 tan pronto recibe el mensaje de cancelación.

Alternativamente, aunque en el estado activo 501, el sistema de cómputo de implementación 201 puede recibir un mensaje salido del sistema de cómputo de implementación 202. Este informa al sistema de cómputo de implementación 201 que el sistema de cómputo 202 no participará más en el patrón de intercambio de mensajes. De acuerdo con lo anterior, la transmisión del mensaje de salida del sistema de cómputo de implementación 202 al sistema de cómputo de implementación 201 resulta en una transición 522 del estado activo 501 al estado de finalización 505.

Alternativamente, mientras en el estado activo 501, el sistema de cómputo de implementación 201 puede recibir un mensaje completado del sistema de cómputo implementación 202. Este informa al sistema de cómputo de incremento 201 que el sistema de cómputo de implementación 202 ha completado el procesamiento relacionado con una solicitud previa. De acuerdo con lo anterior la transmisión del mensaje completado del sistema de cómputo de implementación 202 al sistema de cómputo de implementación 201 resulta en una transición 523 del estado activo 501 al estado completado 503.

Alternativamente, mientras en el estado activo 501, el sistema de cómputo de implementación 201 puede recibir un mensaje de fallido del sistema de cómputo implementación 202. Este informa al sistema de cómputo de implementación 201 que el sistema de cómputo de implementación 202 ha fallado en el estado activo 501. De acuerdo con lo anterior, la transmisión del mensaje de fallido del sistema de cómputo de implementación 202 al sistema de cómputo e implementación 201 resulten una transición 526 del estado activo 501 al estado fallido 507.

Mientras está en el estado de cancelación 502, el sistema de cómputo de implementación 201 puede recibir un mensaje de cancelado del sistema de cómputo de implementación 202. Este informa al sistema de cómputo de implementación 201 que es sistema de cómputo de implementación 202 reconoce que la operación solicitada previamente ha sido cancelada en la solicitud del sistema de cómputo de implementación 201. De acuerdo con lo anterior, la transacción del mensaje cancelado del sistema de cómputo de implementación 202 al sistema de cómputo de implementación 201 resulten una transición 501 del estado de cancelación 502 al estado de finalización 505.

Mientras está en el estado concretado 503, el sistema de cómputo de implementación 201 puede transmitir un mensaje de cerrado al sistema de cómputo de implementación 202. Este informa al sistema de cómputo de implementación 202 que la solicitud previa fue exitosa. De acuerdo con lo anterior, la transmisión del mensaje cerrado del sistema de cómputo de implementación 201 al sistema de cómputo de implementación 202 resultan una transición 512 del estado completado 503 al estado de cerrado 504.

Mientras está en el estado de cerrado 504, el sistema de cómputo de implementación 201 puede recibir un mensaje de cerrado del sistema de cómputo de implementación 202. Este informa al sistema de cómputo de implementación 201 que el sistema de cómputo de implementación 202 ha finalizado exitosamente la operación. De acuerdo con lo anterior, la transmisión del mensaje cerrado del sistema de cómputo de implementación 202 al sistema de cómputo y de implementación 201 resulta en una transición 524 del estado de cerrado 504 al estado de finalización 505.

También mientras está en el estado completado 503 el sistema de cómputo de implementación 201 puede transmitir un mensaje de compensación al sistema de cómputo de implementación 202. Este informa la sistema de cómputo de implementación 202 que el trabajo está siendo hecho debería ser desecho a un grado razonable. De acuerdo con lo anterior, la transmisión del mensaje de compensación del sistema de cómputo de implementación 201 al sistema de cómputo de implementación 202 resultan en una transición 513 del estado completado 503 al estado de compensación 506.

Mientras está en el estado de compensación 506, el sistema de cómputo de implementación 201 puede recibir un mensaje compensado del sistema de cómputo de implementación 202. Este informa al sistema de cómputo de implementación 202 que la acción de compensación fue exitosa. De acuerdo con lo anterior, la transmisión del mensaje compensado del sistema de cómputo de la implementación 202 al sistema de cómputo de implementación 201 resulta en una transición 525 del estado de compensación 506 al estado de finalización 505.

Alternativamente, mientras está en el estado de compensación 506, el sistema de cómputo de implementación 201 puede recibir un mensaje de fallido del sistema de cómputo de implementación 202. Este informa al sistema de cómputo de implementación 201 que el sistema de cómputo de implementación 202 a fallado en el estado de compensación 506. De acuerdo con lo anterior, la transmisión del mensaje de fallido del sistema de cómputo de implementación 202 al sistema de cómputo de implementación 201 resulta en una transición 527 del estado de compensación 506 al estado de fallido 507.

Mientras está en el estado fallido 507, el sistema de cómputo de implementación 201 puede transmitir un mensaje del olvido al sistema de cómputo de implementación 202. Este informa al sistema de cómputo de implementación 202 que el sistema de cómputo de implementación 201 está consciente de la falla. De acuerdo con lo anterior, la transmisión del mensaje de olvido del sistema de cómputo de implementación 201 al sistema de cómputo de implementación 202 resulta en una transición 514 del estado de fallido 507 al estado de finalización 505.

La figura 6 ilustra un diagrama de flujo de un método 600 para el sistema de cómputo de generación de códigos 301 para generar el código automáticamente el cual ensaya las capacidades del sistema de cómputo de prueba 401 para usar una aplicación del patrón del intercambio de mensajes para enganchar en transacciones de mensajes siguiendo un patrón de intercambio de mensajes. El código permite la simulación sin requerir que el sistema de cómputo de prueba 401 se enganche en la transacción de mensajes con otras organizaciones o aún en algunos casos sin requerir la comunicación con otros sistemas de cómputo. El método 600 puede ser descrito con referencia

frecuente al ejemplo específico de la definición del patrón de intercambio de mensajes de la figura 5.

El sistema de cómputo de generación de códigos accesa una definición de patrón de intercambio de mensajes (acto 601) tal como la definición del patrón de intercambio de mensajes descrito anteriormente. El sistema de cómputo de generación de códigos lleva a cabo entonces una etapa funcional orientada a resultados para genera de manera automática el código de simulación de patrón de intercambio de mensajes usando la definición del patrón de intercambio de mensajes (etapa 610). La etapa 610 puede incluir cualesquier acto específicos correspondientes que lleven a cabo este resultado. Sin embargo, en la modalidad ilustrada en la etapa 610 incluye los actos 611 y 612 correspondientes.

Específicamente, para cada estado en el cual la definición de patrón de intercambio de mensajes permite que un mensaje de transmisión válido se ha transmitido por el sistema de cómputo de prueba, el sistema de cómputo de generación de códigos automáticamente genera un código que por lo menos simula la transmisión del mensaje de transmisión válido (acto 611). Por ejemplo,

considere el estado activo 501 de la figura 5. El estado activo 501 permite que un mensaje válido sea transmitido, un mensaje de cancelación. Se ha indicado que es un cambio del 3% que el mensaje de cancelación será transmitido. Este porcentaje puede ser específico por la definición del patrón de intercambio de mensajes y/o puede ser especificada por el usuario o por el sistema de cómputo de generación de códigos. El porcentaje asociado con esta transmisión puede cambiar dinámicamente durante el tiempo durante el proceso de ensayo. Para simular la transmisión condicional del mensaje de cancelación el sistema de cómputo de generación de códigos puede generar el código que sigue el siguiente documento de pseudo código con el número de línea agregado para facilidad de la explicación posterior:

```
1.  main
2.  {
3.      Do State501( )
4.      {
5.          Flip a Coin
6.          if Coin is in 3% range
7.              Send(Cancel)
```

```
8.          Go to DoState502
9.      end if
10.     }
11.     .
12.     .
13.     .
14. }
```

En este pseudo código un programa principal es creado con subtítulo en la línea 1, su corchete de apertura en la línea 2, y su corchete de cierre en la línea 14. Entonces un sub-rutina es generado llamada Do State 501. La línea 3 identifica la sub-rutina la cual está entre corchetes usando un corchete abierto en la línea 4 y un corchete de cierre en la línea 10. El código es entonces generado para generar un valor pseudo aleatorio (ver línea 5). El código condicional es entonces generado de modo que si es apropiado dado que el valor aleatorio (ver línea 6), la transmisión del mensaje da cancelación es por lo menos simulada (ver línea 7) y el código es generado para avanzar la ejecución al siguiente estado (ver línea 8) para así hacer la transición al siguiente estado apropiado.

Regresando de nuevo a la figura 6, para cada estado en el cual la definición del patrón de intercambio de mensajes permite que un mensaje de recepción válido sea recibido, el sistema de cómputo de generación de códigos genera el código que simula la recepción del mensaje de recepción válido, y que las transiciones a otro código que represente el estado a la transición. Por ejemplo el estado activo 501 permite la recepción de tres mensajes válidos; un mensaje de salida que tiene una probabilidad de 10% de ocurrencia, un mensaje de completado que tiene un 75% de oportunidad de ocurrencia y un mensaje de fallido que tiene un 2% de probabilidad de ocurrencia. Esto es simulado en el pseudo código de ejemplo agregando afirmaciones condicionales adicionales en la misma subrutina DoState 501 como sigue:

```
1.  main
2.  {
3.      Do State501( )
4.      {
5.          Flip a Coin
6.          if Coin is in the appropriate 3% range
```

```
7.          Send(Cancel)
8.          Go to DoState502
9.      end if
10.     if Coin is in the appropriate 10% range
11.         Receive(Exited)
12.         Go to DoState505
13.     end if
14.     if Coin is in the appropriate 75% range
15.         Receive(Completed)
16.         Go to DoState503
17.     end if
18.     if Coin is in the appropriate 2% range
19.         Receive(Faulted)
20.         Go to DoState507
21.     end if
22. }
23. .
24. .
25. .
26. }
```

Note que la adición del pseudo código ejecutado condicionalmente para simular la recepción del mensaje de

salida con la transición del estado apropiado al estado 505 (ver líneas 10 a 13), simula la recepción del mensaje completado con el estado apropiado de transición al estado 503 (ver líneas 14-17), y simula la recepción de mensaje de fallido con el estado apropiado de transición al estado 507 (ver líneas 18 a 21).

En adición, la transmisión o recepción de mensajes inválidos puede ser simulada por medio de asignar una probabilidad positiva de ocurrencia a la transmisión o recepción de ciertos mensajes inválidos. Estos mensajes inválidos pueden no ser especificados por la definición del patrón de intercambio de mensajes por que ellos no siguen el patrón de intercambio de mensajes. En su lugar, ellos pueden ser ingresados por un usuario o por el sistema de cómputo de generación de códigos, por ejemplo, mientras está en el estado activo 501 un mensaje inválido puede ser transmitido como se indicó por la flecha 541 (que tiene un 7% de probabilidades de ocurrencia). también un mensaje inválido puede ser recibido como se indicó con la flecha 542 (que tiene un 3% de probabilidad de ocurrencia). Simular la recepción y la transición de un mensaje inválido permite la simulación para probar la

posibilidad muy real de tener que manejar con la no conformidad al patrón de intercambio de mensajes. Un código similar puede ser generado automáticamente para estos mensajes inválidos también, el siguiente pseudo código extiende el ejemplo anterior a generar un código que simule condicionalmente la transmisión (ver líneas 25 a 27) y la recepción (ver líneas 22 a 24) de mensajes inválidos.

```
1.  main
2.  {
3.      Do State501( )
4.      {
5.          Flip a Coin
6.          if Coin is in the appropriate 3% range
7.              Send(Cancel)
8.              Go to DoState502
9.          end if
10.         if Coin is in the appropriate 10% range
11.             Receive(Exited)
12.             Go to DoState505
13.         end if
14.         if Coin is in the appropriate 75% range
```

```
15.          Receive(Completed)
16.          Go to DoState503
17.      end if
18.      if Coin is in the appropriate 2% range
19.          Receive(Faulted)
20.          Go to DoState507
21.      end if
22.      if Coin is in the appropriate 3% range
23.          Receive(Invalid Message)
24.      end if
25.      if Coin is in the appropriate 7% range
26.          Send(Invalid Message)
27.      end if
28.  }
29.  .
30.  .
31.  .
32. }
```

Este proceso de generación puede también ser llevado a cabo para generar sub-rutinas para otros estados tan bien (por ejemplo, los estados 502 a 507), que también pueden tener sus propias probabilidades de ramificación. El

proceso de generación de código involucra generar un programa principal (por ejemplo, el programa "main"), generar una sub-rutina para cada estado, generando un número aleatorio para cada estado en el cual múltiples eventos pueden ocurrir, y luego condicionalmente llevar a cabo el evento con base en el número aleatorio. Como esto es un altamente procedural, involucra poco en la manera de llamadas de juicio, el proceso de generación de código puede ser altamente automatizado. Cualquier intervención de usuario si hay alguna, seria trivial. Por ejemplo, el usuario puede requerir estimar la probabilidades de ocurrencia para cada tipo de mensaje válido e indicar si o no la recepción o transmisión de mensajes inválidos puede ocurrir (junto con probabilidades asociadas) sin embargo, el usuario no requeriría realmente de bosquejar el código mismo. De acuerdo con lo anterior, el código en es generado más fácilmente.

Adicionalmente las funciones "send" y "receive" pueden ser generadas para simular simplemente la transmisión o recepción de mensajes a medida que ellos ocurren durante el despliegue real De acuerdo con lo anterior, no hay necesidad de descansar en la cooperación de otras organizaciones que en últimas estarán involucradas con un

patrón de intercambio de mensajes para poder ensayar la aplicación de intercambio de mensajes. En una modalidad, la transmisión y recepción mismas pueden ser simuladas de modo que la simulación resulte en ningún tráfico de red real preservando así los recursos de red.

La presente invención puede ser construida en otras formas específicas sin salir de su espíritu o características esenciales las modalidades descritas deben considerarse en todo los aspectos solamente como ilustrativa y no como restrictivas. El alcance de la invención, es por lo tanto, indicado por las reivindicaciones adjuntas en vez de la descripción anterior. Todos los cambios, que surjan dentro del significado y rango de equivalencia de las reivindicaciones deben ser considerados dentro de su alcance. Lo que se reivindica y desea asegurarse por una patente es:

REIVINDICACIONES

1. En un sistema de cómputo de generación de códigos que incluye uno o máspreciadores capases de ejecutar instrucciones ejecutables por computador en una memoria del sistema, un método para que el sistemas de cómputo de generación de códigos genera de manera automática un código que pruebe las capacidades de un sistema de cómputo de prueba para usar una aplicación de patrón de intercambio de mensaje para enganchar o ejecutar transacciones de mensaje siguiendo un patrón de intercambio de mensajes el método comprende lo siguiente:

Un acto de accesar una definición de patrón de intercambio de mensajes que define una pluralidad de estados, la definición de patrón de intercambio de mensaje tradicionalmente tiene para cada una de la pluralidad de estados, una indicación en uno o más mensajes válidos que conforman el patrón de intercambio de mensajes para ese estado, una indicación de qué sistema de cómputo puede transmitir cada mensaje válido dado el estado, y una indicación de transición de

estado para por lo menos que algunos de los mensajes válidos identifique una de la pluralidad de estados a la transición en que debería ocurrir el mensaje válido;

Para cada estado en el cual la definición del intercambio de mensajes permite que un mensaje de transmisión válido ser transmitido por el sistema de cómputo de prueba, un acto de generar un código que por lo menos simule la transmisión del mensaje de transmisión válido y que las transiciones a otro código que represente el estado a la transición en que debería el mensaje de transmisión válido ser transmitido; y

Para cada estado en el cual la definición del patrón de intercambio de mensajes permite que un mensaje de recepción válido sea recibido un acto de generar un código que simulen la recepción del mensaje de recepción válido, y que las transiciones a otro código que represente el estado de la transición en la cual el mensaje de transición válido es recibido.

2. Un método De acuerdo con la reivindicación 1 en donde la definición del patrón de intercambio de mensajes indica que cuando en un estado particular, cualquiera de la pluralidad de mensajes de transmisión válidos pueda ser transmitido.
3. Un método de acuerdo con la reivindicación 2, en donde la definición del intercambio de mensajes indica para cada una de la pluralidad de mensajes de transmisión válido para el estado particular hubo un porcentaje de probabilidades de que cada uno de la pluralidad de mensajes de transmisión válido será transmitido dado el estado particular, en donde el método comprende adicionalmente generar un código de manera automáticamente para por lo menos simular cada uno de la pluralidad de mensajes de transmisión válidos y llevar a cabo las transiciones a los estados apropiados dada la transmisión.
4. Un método de acuerdo con la reivindicación 3 que comprende adicionalmente lo siguiente:

un acto de generar un código que genere un valor pseudo aleatorio y seleccione una de pluralidad de mensajes de transmisión válido para la transmisión con base en el valor pseudo aleatorio y en un porcentaje de probabilidad.

5. Un método de acuerdo con la reivindicación 1, que comprende adicionalmente los siguiente para por lo menos un estado:

un acto de generar un código que por lo menos simule la transmisión de un mensaje de transmisión inválido.

6. Un método de acuerdo con la reivindicación 5, en donde hay también indicado un porcentaje de probabilidad de que en los mensajes de transmisión invadidos sean transmitidos dado el estado particular en donde el método comprende adicionalmente un acto de generar un código que genera un valor pseudo aleatorio y selecciona uno de los mensajes de transmisión inválidos para la transmisión con base en el valor pseudo aleatorio y en el porcentaje de probabilidad.

7. Un método de acuerdo con la reivindicación 1, en donde la definición de intercambio de mensajes indica que cuando en un estado particular, cualquiera de una pluralidad de mensajes de recepción válidos sea recibido.
8. Un método de acuerdo con la reivindicación 7, en donde la definición de patrón de intercambio de mensajes indica para cada uno de la pluralidad de mensajes de recepción válidos para el estado particular, un porcentaje de probabilidad de que cada una de la pluralidad de mensajes de recepción válidos será recibido dado el estado particular, en donde el método comprende adicionalmente generar de manera automática un código para simular cada una de la pluralidad de mensaje de recepción válidos y llevar a cabo las transiciones a los estados apropiados dada la transmisión.
9. Un método de acuerdo con la reivindicación 8, que comprende adicionalmente lo siguiente:

Un acto de generar un código que genera un valor pseudo aleatorio y selecciona uno entre la pluralidad de mensajes de recepción válido para la recepción simulada con base en el valor pseudo aleatorio y en el porcentaje de probabilidad.

10. Un método de acuerdo con la reivindicación 1, que comprende adicionalmente los siguiente para por lo menos un estado:

un acto de generar un código que simule la recepción de un mensaje de transmisión inválido.

11. Un método de acuerdo con la reivindicación 10, en donde se ha indicado también un porcentaje de probabilidad de que el mensaje de recepción inválido será recibido dado el estado particular, en donde el método comprende adicionalmente un acto de generar un código que genera un valor pseudo aleatorio y selecciona uno de los mensajes recibidos inválido para simular la recepción con basé al valor pseudo aleatorio y en el porcentaje de probabilidad.

12. Un método de acuerdo con la reivindicación 1, en donde la definición del patrón de intercambio de mensajes es definida usando el estándar de lenguaje de descripción de servicios de red (WSDL).
13. Un método de acuerdo con la reivindicación 1, en donde la definición del patrón de intercambio de mensajes define adicionalmente las políticas de temporización que deben ser impuestas cuando está en un estado particular.
14. Un producto de programa de computador para uso en un sistema de generación de códigos que incluye uno o más procesadores capaces de ejecutar instrucciones ejecutables por computadora en la memoria del sistema, el producto de programa de computador para llevar a cabo un método para el sistema de cómputo de generación de códigos para generar de manera automática un código que se pruebe las capacidades de un sistema de cómputo de prueba para usar una aplicación de patrón de intercambio de mensaje para llevar a cabo transacciones de

mensaje siguiendo un patrón de intercambio de mensajes, el producto de programa de cómputo comprende uno o más medios legibles por computadora que, cuando son ejecutados por uno o más procesadores el sistema de cómputo de generación de códigos, causa que el sistema de cómputo de generación de códigos lleve acabo el método, el método comprende lo siguiente:

un acto de accesar una definición de patrón de intercambio de mensaje que define una pluralidad de esperados, en la definición de patrón de intercambio de mensajes tiene adicionalmente para cada una de la pluralidad de estados, una invitación de uno más mensajes válidos que se conforman al patrón de intercambio de mensajes para ese estado una indicación de cual sistema de cómputo puede transmitir cada mensaje válido dado del estado, y una indicación de transición de estado para por lo menos algunos de los mensajes válidos identifique uno de la pluralidad de estados para la transición que debería ocurrir en el mensaje válido;

Para cada estado en el cual la definición del patrón de intercambio de mensaje permite que un mensaje de transmisión válido sea transmitido por el sistema de cómputo de prueba, un acto de generar un código que por lo menos simule la transmisión del mensaje de transición válido y que las transiciones a otro código que represente el estado de la transición en que debería el mensaje de transmisión válido ser transmitido; y

Para cada estado en el cual la definición del patrón de intercambio de mensajes permite que un mensaje de recepción válido sea recibido, un acto de generar un código que simule la recepción del mensaje de recepción válido, y las transiciones a otro código que represente el estado de la transición a la cual el mensaje de recepción válido es recibido.

15. Un producto de programa de computador acuerdo con la reivindicación 14, en donde la definición del patrón de intercambio de mensajes indica que cuando está en un estado particular, cualquiera

de la pluralidad de mensajes de transmisión válido sea transmitido.

16. Un producto de programa de computador de acuerdo con la reivindicación 15, en donde la definición del patrón de intercambio de mensajes indica para cada una de la pluralidad de mensajes de transmisión válidos para el estado particular, un porcentaje de probabilidad de que cada uno de la pluralidad de mensajes de transmisión válido será transmitido dado el estado particular, en donde el método comprende adicionalmente generar automáticamente un código para que por lo menos simule cada una de la pluralidad de mensajes de transmisión válidos y lleve a cabo las transiciones de estado apropiadas dada la transmisión.

17. Un producto de programa de computador de acuerdo con la reivindicación 16, que comprende adicionalmente lo siguiente:

Un acto de generar un código que genere un valor pseudo aleatorio y seleccionar una de la

pluralidad de mensajes de transmisión válido para la transmisión con base en el valor pseudo aleatoria y en el porcentaje de probabilidad.

18. Un producto de programa de computador de acuerdo con la reivindicación 14, que comprende adicionalmente lo siguiente para por lo menos un estado:

Un acto de genera un código que por lo menos simule la transmisión de un mensaje de transmisión inválido.

19. Un producto de programa de computador de acuerdo con la reivindicación 18, en donde se indique también un porcentaje de probabilidad de que los mensajes de transmisión inválidos serán transmitidos dado el estado particular, en donde el método comprende adicionalmente un acto de generar un código que genere un valor pseudo aleatorio y seleccione uno de los mensajes de transmisión inválidos para la transmisión con base en el valor pseudo aleatorio y en el porcentaje de probabilidad.

20 Un producto de programa de computador de acuerdo con la reivindicación 14, en donde la definición del patrón de intercambio de mensajes indica que cuando está en un estado particular cualquiera de la pluralidad de mensajes de recepción válidos puede ser recibido.

21 Un producto de programa de computador de acuerdo con la reivindicación 20, en donde la definición del patrón de intercambio de mensajes indica para cada uno de la pluralidad de mensajes de recepción válidos para el estado particular, un porcentaje de probabilidad de que cada una de la pluralidad de mensajes de recepción válidos será recibido dado el estado particular, en donde el método adicionalmente comprende en generar automáticamente un código para simular cada uno de la pluralidad de mensajes de recepción válidos y llevar a cabo las transiciones de estado apropiadas dada la transmisión.

22. Un producto de programa de computador de acuerdo con la reivindicación 21, que comprende adicionalmente lo siguiente:

Un acto de generar un código que genere un valor pseudo aleatorio y seleccione uno entre la pluralidad de mensajes de recepción válidos para simular la recepción con base en el valor pseudo aleatorio y en el porcentaje de probabilidad.

23. Un producto de programa de computador de acuerdo con la reivindicación 14, que comprende adicionalmente lo siguiente para por lo menos un estado:

Un acto de generar un código que simule la recepción de un mensaje de transmisión inválido.

24 Un producto de programa de computador de acuerdo con la reivindicación 23, hay también indicado un porcentaje de probabilidad de que el mensaje de recepción inválido será recibido dado el estado particular, en donde el método comprende adicionalmente un acto de generar un código que

genere un valor pseudo aleatorio y seleccione uno de los mensajes recibidos inválidos para simular la recepción con base en el valor pseudo aleatorio y en el porcentaje de probabilidad.

25. Un producto de programa de computador de acuerdo con la reivindicación 14, en donde la definición del patrón de intercambio de mensajes es definida usando el estándar de lenguaje de descripción de servicios de red (WSDL).

26. Un producto de programa de computador de acuerdo con la reivindicación 14, en donde la definición del patrón de intercambio de mensajes define adicionalmente las políticas de temporización que deben ser impuestas cuando está en un estado particular.

27 Un producto de programa de computador de acuerdo con la reivindicación 14, en donde el uno o más medios legibles por computador son medios físicos.

28. Un producto de programa de computador de acuerdo con la reivindicación 27, en donde el uno o más medios legibles por computador incluyen una memoria de sistemas.

29. Un producto de programa de computador de acuerdo con la reivindicación 27, en donde el uno o más medios legibles por computador incluye una memoria persistente.

30. Un producto de programa de computador de acuerdo con la reivindicación 29, en donde la memoria persistente.

Es un disco magnético.

31. En un sistema de cómputo de generación de códigos que incluye uno o más procesadores capaces de ejecutar instrucciones ejecutables por computadora en una memoria del sistema un método para que el sistema de cómputo de generación de códigos genere de manera automática un código que verifique o pruebe las capacidades de un sistema de cómputo de prueba para usar una aplicación de

patrón de intercambio de mensajes para llevar a cabo transacciones de mensajes siguiendo un patrón de intercambio de mensajes, el método comprende los siguiente:

Un acto de accesar una definición de patrón de intercambio de mensajes que define una pluralidad de estados, la definición de patrón de intercambio de mensajes tiene adicionalmente para cada una de la pluralidad de estados, una indicación de uno o más mensajes válidos que se conformen al patrón de intercambio de mensajes para ese estado, una indicación de cual sistema de cómputo puede transmitir cada mensaje válido dado el estado y una indicación de transición de estado para por lo menos algunos de los mensajes válidos para que por lo menos alguno de los mensajes válidos identifique una de la pluralidad de estados para la transición que debería ocurrir para el mensaje válido; y

Una etapa para genera de manera automática un código de simulación de patrón de intercambio

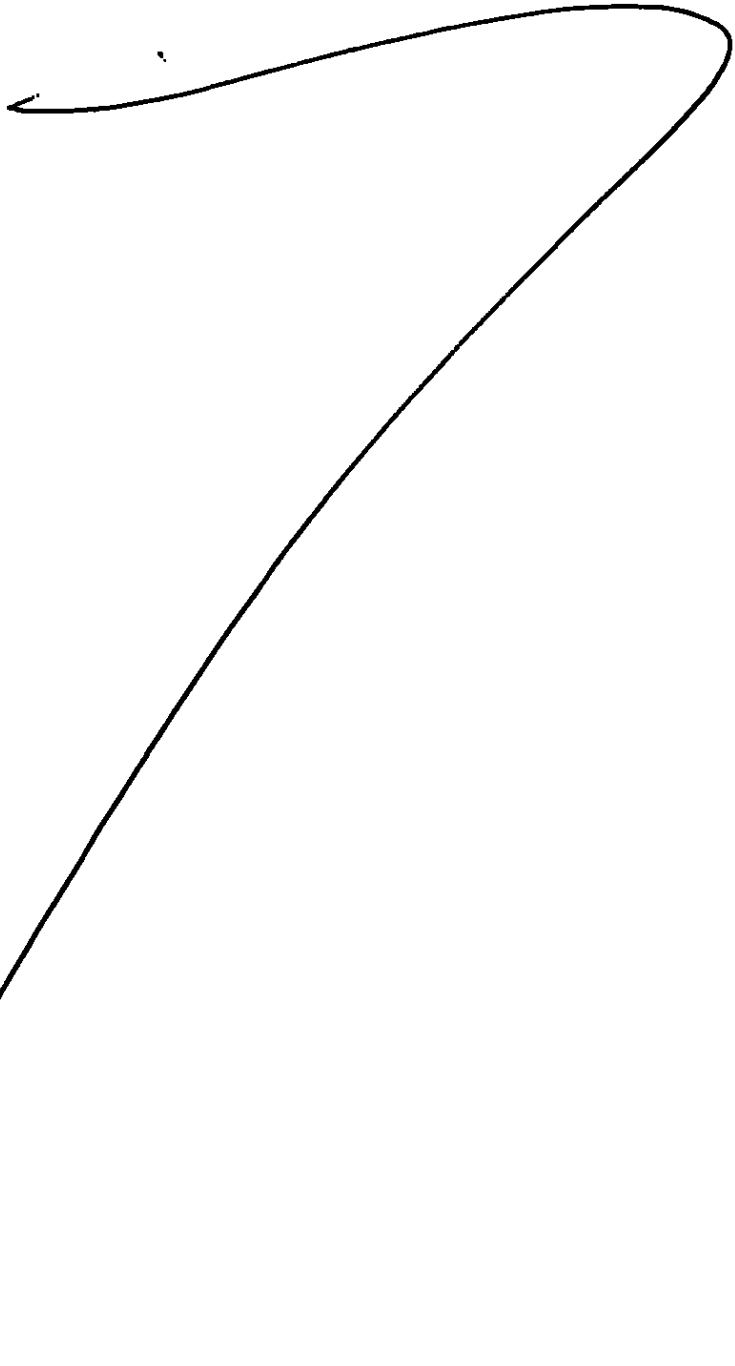
mensajes, usando la definición del patrón de intercambio de mensajes.

32. Un método de acuerdo con la reivindicación 31, en donde la etapa para generar automáticamente un código de simulación de patrón de intercambio de mensajes usando la definición de intercambio de mensajes comprende lo siguiente:

Para cada estado en el cual la definición del patrón de intercambio de mensajes permite que un mensaje de transmisión válido sea transmitido por el sistema de cómputo de prueba, un acto de generar un código que por lo menos simule la transmisión del mensaje de transmisión válido, y las transiciones a otro código que represente el estado en la transición que debería el mensaje de transmisión ser transmitido; y.

Para cada estado en el cual la definición del patrón de intercambio de mensaje permite que un mensaje de recepción válido sea recibido, un acto de generar un código que simule la recepción del mensaje de recepción válido, y que las

transiciones a otro código que represente el estado de transición en le cual el mensaje de recepción de vería ser recibido.



RESUMEN

Los mecanismos para generar automáticamente un código que pruebe las capacidades de un sistema de cómputo de prueba para simular un patrón de intercambio de mensaje. El sistema de cómputo de generación de códigos usa una definición de patrón de intercambio de mensajes para generar el código de simulación. Para cada estado en el cual la definición del patrón de intercambio de mensajes permite que mensajes válidos sean transmitidos, el código es generado para ese estado que por lo menos simule la transmisión de un mensaje de transmisión válido. Para cada estado en el cual la definición de patrón de intercambio de mensajes permite que mensajes válidos, el código es generado para ese estado de modo que simule la recepción de un mensaje de recepción válido. Si la transmisión o recepción de los mensajes causa que un estado de transición ocurra, el código es generado de modo que se cause la transición al estado apropiado en el patrón de intercambio de mensajes.

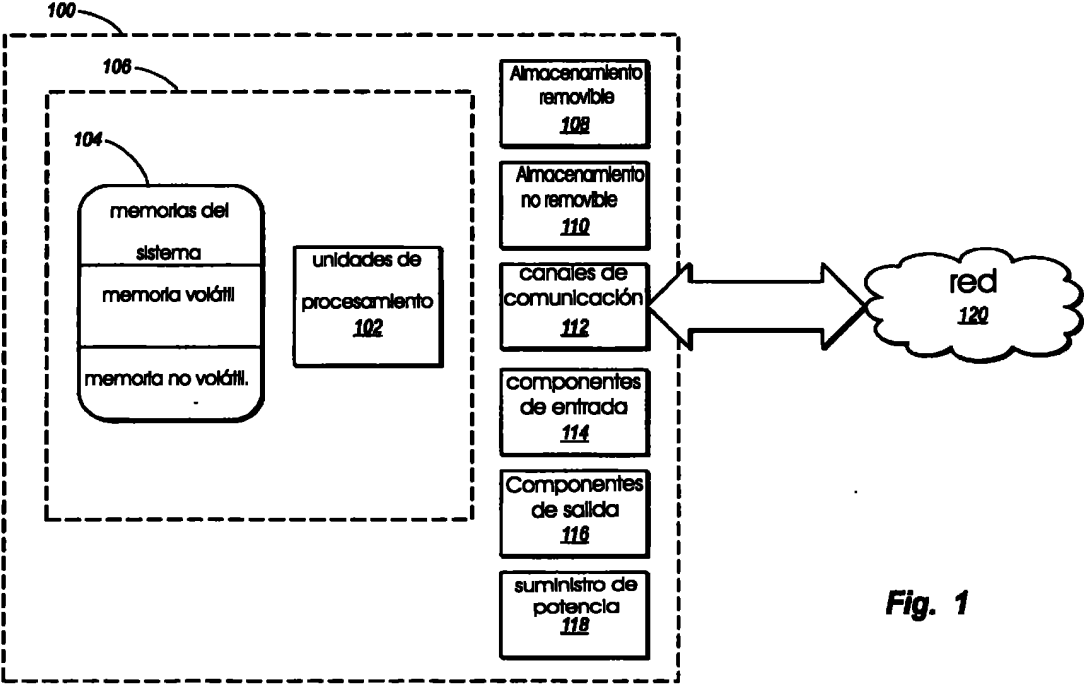


Fig. 1

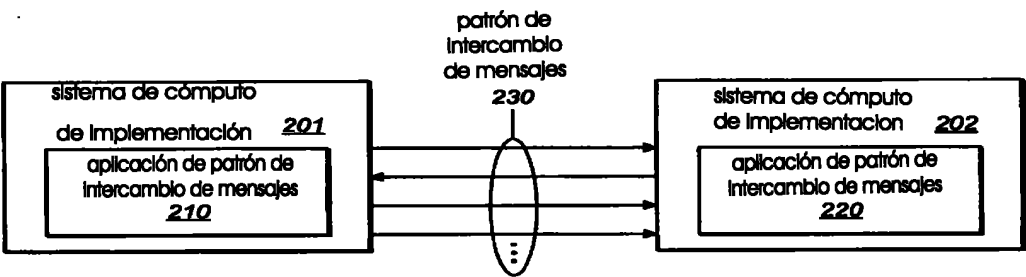


Fig. 2

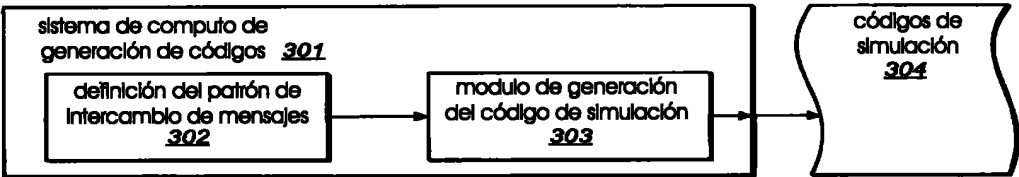


Fig. 3

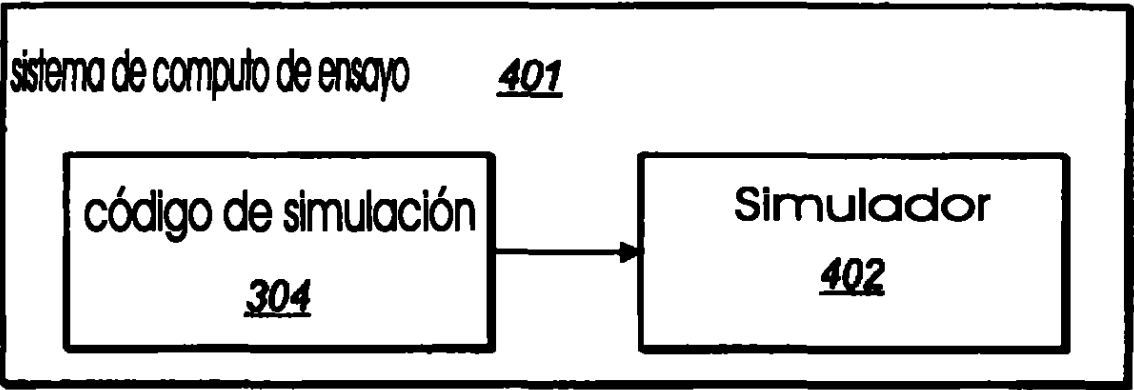


Fig. 4

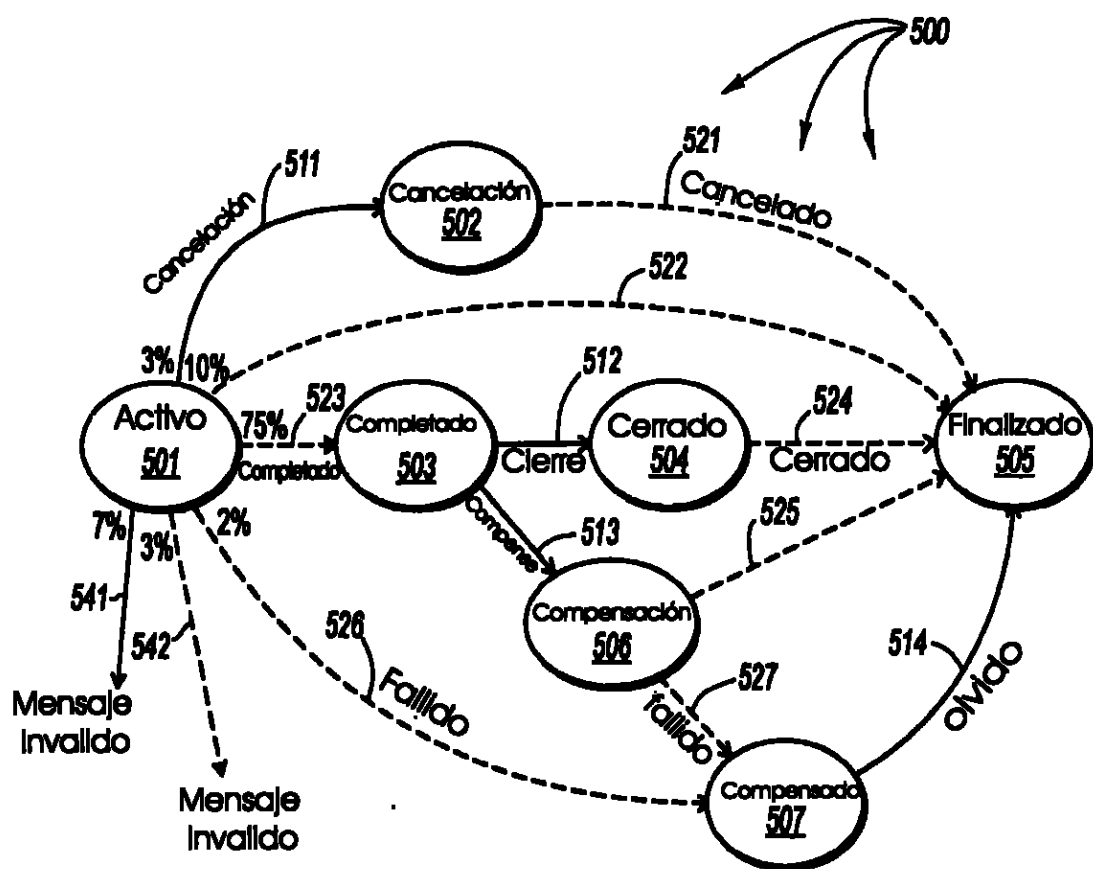
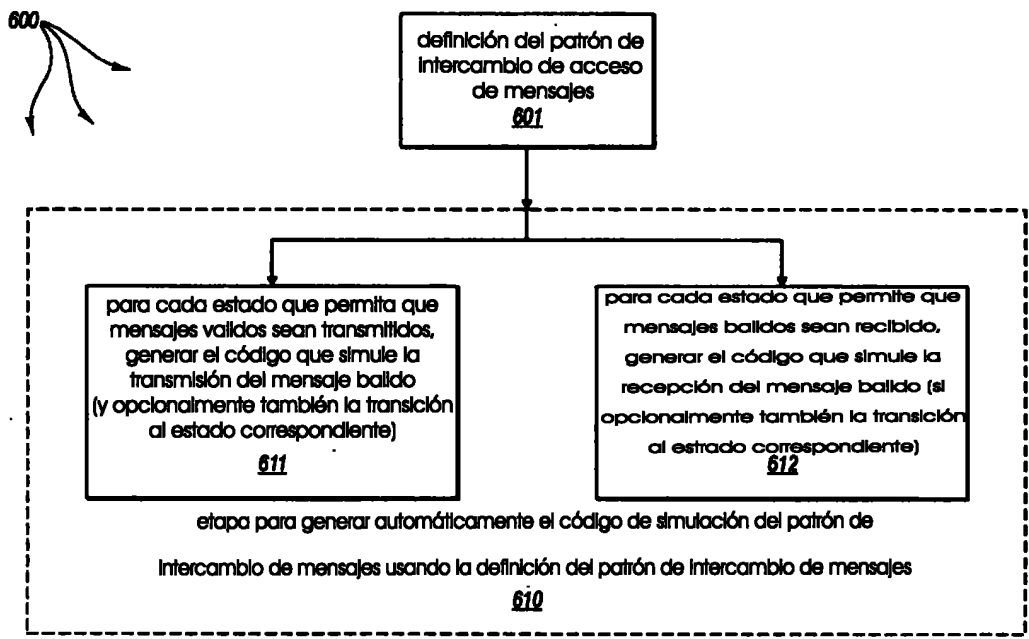


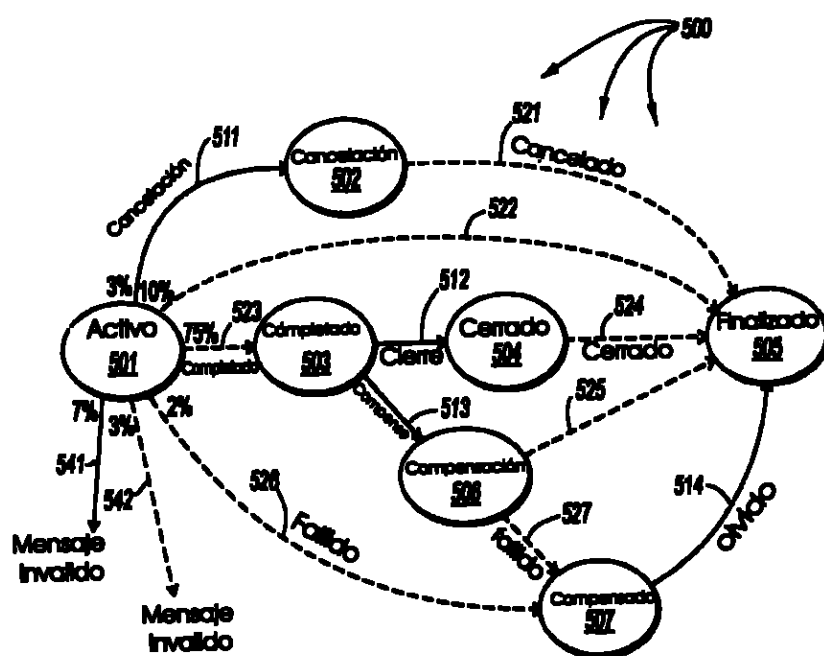
Fig. 5



Fia. 6

ANTE FINAL
12x12

109



115

110
446

REQUISITOS NECESARIOS PARA PRESENTACION Y AGILIZACION DEL TRAMITE

	USUARIO	RADICADOR
PATENTE DE INVENCION	()	()
MODELO DE UTILIDAD	()	()
- Indicación de que se solicita la concesión de una patente.	()	()
- Datos de identificación del solicitante o de la persona que se presenta la solicitud.	()	()
- Descripción de la invención.	()	()
- Dibujos de ser estos pertinentes.	()	()
- Comprobante de Pago de las tasas establecidas.	()	()
COMPLETA ()	INCOMPLETA ()	()
DISEÑO INDUSTRIAL ()		
- Indicación de que se solicita el registro de Diseño Industrial	()	()
- Datos de identificación del solicitante o de la persona que presenta la solicitud.	()	()
- Representación gráfica o fotográfica del Diseño industrial o muestra del Material que incorpora el diseño.	()	()
- Comprobante de pago de las tasas establecidas	()	()
COMPLETA ()	INCOMPLETA ()	()
ESQUEMA DE TRAZADO ()		
- Indicación de que solicita el registro de un Esquema de trazado.	()	()
- Datos de identificación del solicitante o de la persona que presenta La solicitud.	()	()
- Representación gráfica del esquema de trazado	()	()
- Comprobante de pago de las tasas establecidas.	()	()
COMPLETA ()	INCOMPLETA ()	()

Firma Responsable:



Fecha:

SUPERINDUSTRIA Y COMERCIO
Radicacion : 05001042 00000000 Folios: 115
Fecha (MM): 2005-01-07 12:14:23
Tramite : 002 PATENTES D 1 REGISTRO/ 411 PRESENTAC
Dependencia: 2020 DIVISION DE NUEVAS CREACIONES

PROTOCOLO

Expediente 05-1042

**SUPERINDUSTRIA Y COMERCIO
SISTEMA DE REGISTRO**

MANTENIMIENTO DE PROTOCOLOS

Numero : 16069

Fecha : 04/08/1995

Conferido: MICROSOFT CORPORATION

Pais : US ESTADOS UNIDOS DE AMERICA

Ciudad : 234 REDMOND, WASHINGTON

Represent: S Sustituir: S Desistir: S

No. Radicacion: 95 34920 Clase: 0 Seev: 0 Seac: 0

scncia	Codigo	Ctr	Nombre
1	77	A	CARDENAS NAVAS DARIO
2	3653		CARDENAS CABALLERO EDUARDO
3	5048		CARDENAS MARTINEZ BERNARDO

Encontrados (1/1) registro(s)

Carrera 13 No. 27-00 Pisos 5, 7 y 10
Conmutador: 382 0840
Fax: 350 5220
E-mail: info sic.gov.co
www.sic.gov.co
Bogotá, D.C., Colombia

Folio 118

2020
Bogotá, D. C.

Doctor(a)
UZ HELENA ADARVE GOMEZ
Apoderado(a)

12072

Oficio No. .

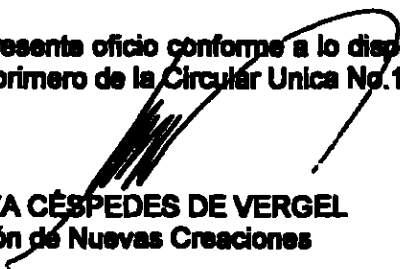
REFERENCIA:	Radicación	05-1042
	Trámite	002
	Evento	001
	Actuación	430
	Folios	001

Como resultado del examen de forma, realizado con fundamento en el artículo 38 de la Decisión 486 de la Comisión de la Comunidad Andina, se le comunica que:

- Allegar copia del documento en que conste la cesión del derecho a la patente otorgado por el inventor al solicitante o a su causante. (art. 26-k, Dec 486/2000).

En cumplimiento del artículo 39 de la Decisión 486, se le requiere para que dentro del término de dos (2) meses siguientes a la fecha de notificación del presente oficio, complete los requisitos anteriormente enunciados. En caso de no dar cumplimiento al presente requerimiento, la solicitud se considerará abandonada y perderá su prelación.

Notifíquese el presente oficio conforme a lo dispuesto en el numeral 5.2, literal e), del capítulo quinto del título primero de la Circular Unica No.10 de 2001.


ALIX CARMENZA CÉSPEDES DE VERGEL
Jefe de la División de Nuevas Creaciones

El anterior requerimiento fue notificado por fijación en lista de fecha 10 FEB. 2005

Se recuerda al interesado que debe presentar, dentro de los dieciséis meses siguientes contados a partir de la fecha de presentación de la solicitud cuya prioridad se invoca, copia de la misma certificada por la autoridad que la expidió y, en los casos en que haya lugar, la traducción simple del capítulo reivindicatorio. El incumplimiento de éste plazo, acarreará la pérdida de la prioridad invocada.

202027

Carrera 13 No. 27-00 Pisos 5, 7 y 10
Conmutador: 382 0840
Fax: 350 5220
E-mail: info@sic.gov.co
www.sic.gov.co
Bogotá, D.C., Colombia