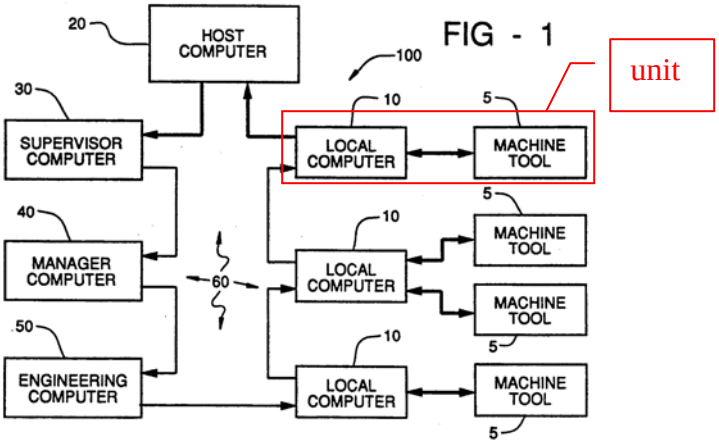
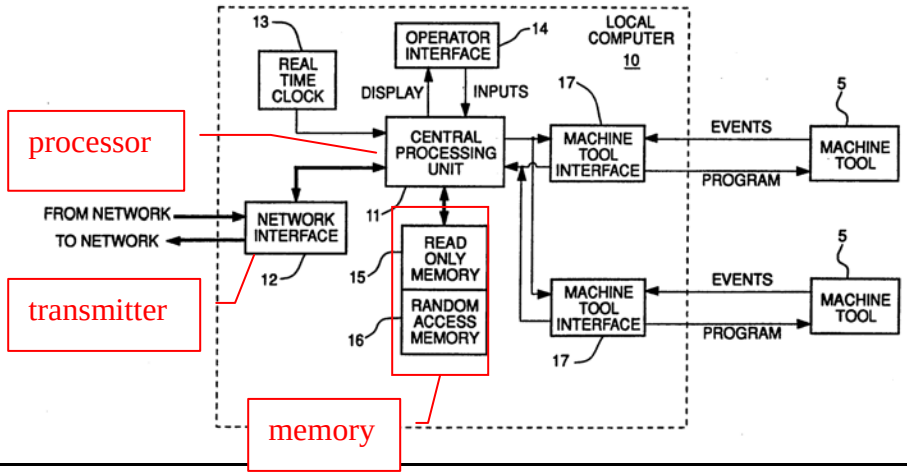
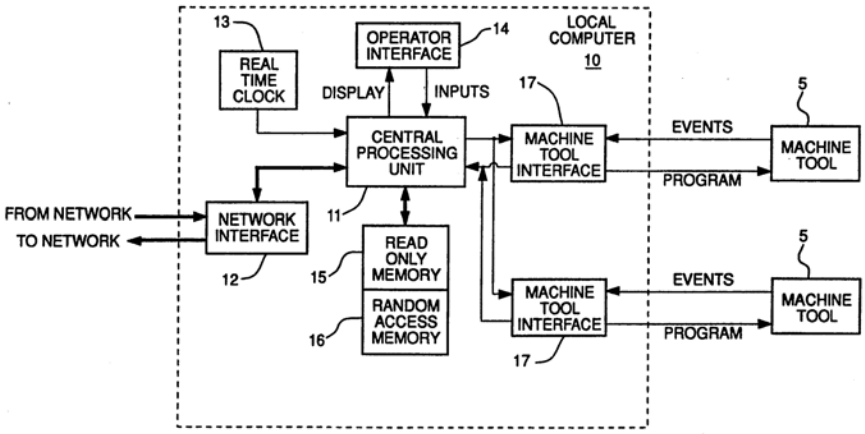
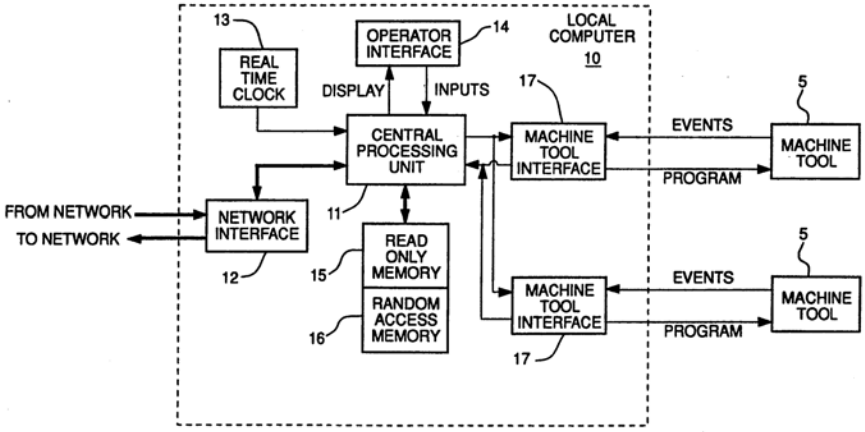
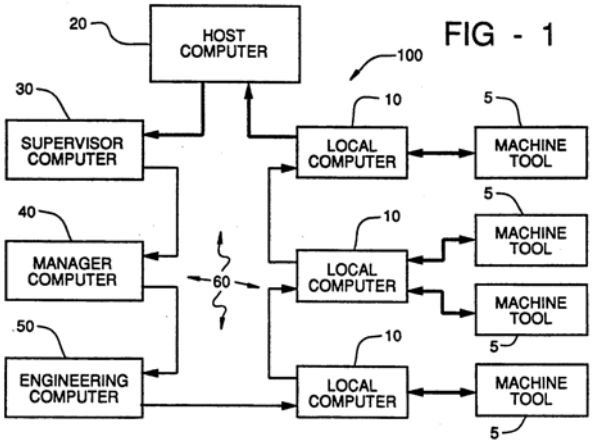
EXHIBIT CC-E

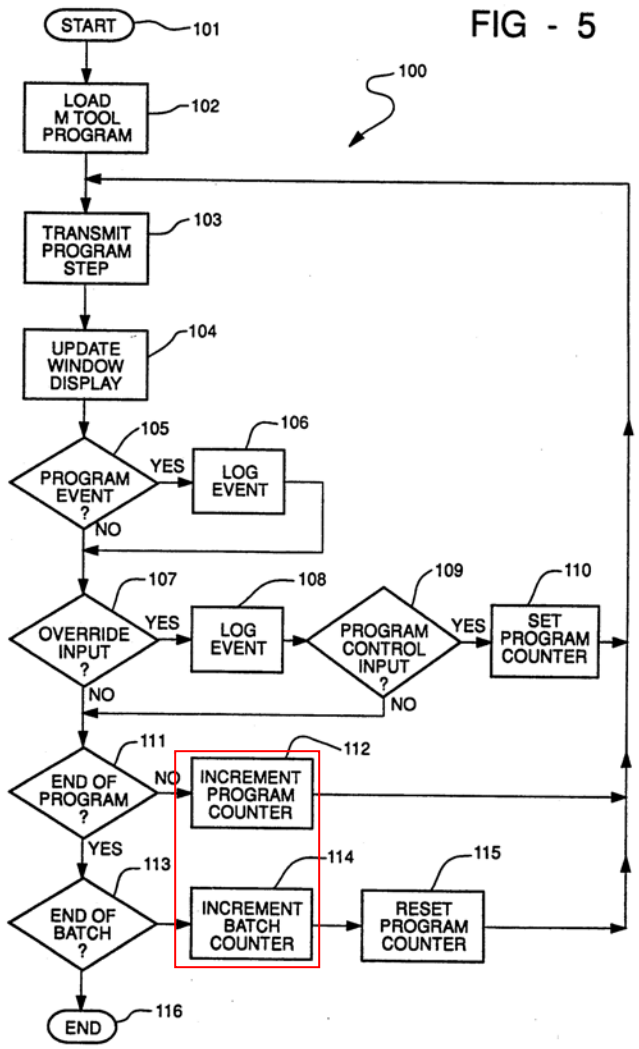
5,291,416 ("HUTCHINS") ANTICIPATES CLAIMS 1, 2, 5-10, 14, 15, 17-22, AND 26-32 OF US PATENT 7,620,565 UNDER 35 U.S.C. §102(E)

Claim 1	Disclosure In Hutchins
A unit, comprising:	Hutchins discloses a unit. Hutchins discloses a local computer (10) connected to a machine tool (5) (collectively, "unit").  (Fig. 1) Hutchins also discloses a computer numerically controlled machine. "This technique may be used with current **computer numerically controlled machines** by revising their control software-firmware to generate the event records and by transmitting same via the serial port to a host computer programmed to accept and store the records." (3:28-32).
a memory;	Hutchins discloses a unit comprising a memory (15, 16) 

	(Fig. 3) "Local computer 10 includes central processing unit 11, network interface 12, real time clock 13, computer user Interface 14, **memory including read only memory 16 and random access read/write memory 16**, and at least one machine tool interface 17." (6:6-10)
a transmitter; and	Hutchins discloses a network interface (12) ("transmitter").  (Fig. 3) "Local computer 10 includes central processing unit 11, **network interface 12**, real time clock 13, computer user Interface 14, memory including read only memory 16 and random access read/write memory 16, and at least one machine tool interface 17." (6:6-10)
a processor, coupled to the memory and to the transmitter, configured to:	Hutchins discloses a processor, coupled to the memory and to the transmitter. 

	(Fig. 3) "Local computer 10 includes **central processing unit 11**, network interface 12, real time clock 13, computer user Interface 14, memory including read only memory 16 and random access read/write memory 16, and at least one machine tool interface 17." (6:6-10)
monitor a product for an occurrence in the product of a trigger event of a predefined plurality of trigger events,	Hutchins discloses that the processor is configured to monitor a product for an occurrence in the product of a trigger event of a predefined plurality of trigger events. Hutchins discloses a local computer connected to a machine tool (in combination, the "product").  (Fig. 1) Hutchins discloses a plurality of trigger events including, but not limited to, input from a touch screen display, input from a keyboard, entering a program event, completion of a part program step, and completion of a part program. "Of local computer 10 and the corresponding machine tool(s) 5, and input devices, i.e., **touch screen keyboard, buttons that produce input signals** as software interrupts that permit the machine tool operator to direct the operation of the system." (6:39-43) "A further subprogram under the run machine tool program checks to determine whether a machine tool part program requires the **logging of an event**. This can occur upon supply of particular machine tool part program steps containing significant events to the machine tool 5 or when the machine tool operator has pressed particular control buttons. As will be described below **an event is logged by identifying the type of event, date and time**

	of occurrence and storing this data in an event buffer within random access memory 16. The logging of an event signals the host communications subprogram that dispatches a message to host computer 20 via the network 60." (13:11-23) "When the machine tool operator presses the cycle start button on the machine tool, this loop begins by transmitting the next **program step** to the machine tool (processing block 103). In the initial operation of the machine tool part program this **next program step** is the **first program step**. As previously disclosed, this process takes place via machine tool interface 17, which communicates with machine tool 5." (14:45-52) "Once this process is complete, then the machine tool operates according to the selected machine tool part program. This process involves a loop that repeats the steps necessary to machine a part until the machine tool **part program is complete**. Completion of the batch of parts involves the execution of the machine tool part program enough times to manufacture the number of parts specified by the batch size." (14:37-44)
increment a counter corresponding to the trigger event upon detection of the occurrence of the trigger event,	Hutchins discloses that the processor is configured to increment a counter corresponding to the trigger event upon detection of the occurrence of the trigger event. Hutchins discloses incrementing a program counter after a program step. "Subprogram 100 next tests to determine whether the just transmitted program step is at the end of the machine tool part program (decision block 111). If this is not the end of the part program then the **program counter is incremented** (processing block 112)." (16:3-7) Hutchins discloses incrementing a counter after the completion of a part program. "Subprogram 100 then tests to determine whether the end of the batch has not been reached then subprogram 100 **increments the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20) Hutchins discloses incrementing the program counter (112) and the batch counter (114) in Figure 5.

	 (Fig. 5)
cause the display of a user interface, configured to probe for information regarding a use of the product,	Hutchins discloses that the processor is configured to cause the display of a user interface, configured to probe for information regarding a use of the product. Hutchins discloses a user interface. "Local computer 10 includes central processing unit 11, network interface 12, real time clock 13, **computer user interface 14**, memory including read only memory 16 and random access read/write memory 16, and at least one machine tool interface 17." (6:6-10) Hutchins discloses displaying a user interface, configured to probe for information regarding a use of the product. "The **computer user interface 14** permits **two way interaction between the machine tool operator and computer** 10. **Computer user interface 14** may include a video display, flat panel display, touch screen,

	keyboard, or buttons as appropriate and serves as the user interface of local computer 10. The **computer user interface 14** includes a display that produces a visual image corresponding to display signals from central processing unit 11 for the user." (6:30-38) "Of local computer 10 and the corresponding machine tool(s) 5, and input devices, i.e., touch screen keyboard, buttons that produce **input signals** as software interrupts that permit the machine tool operator to **direct the operation of the system**." (6:39-43) Hutchins discloses a user interface to permit a machine tool operator to record events that include information about the use of the local computer and machine tool. "The other type of events reported include operator initiated events at the machine tool. These machine tool operator initiated events include, but are not limited to: **down-loading a machine tool part program to the machine tool; setting the batch size; beginning or ending the operation cycle of the machine tool part program; skipping or deleting operations such as may occur when reworking a work piece; editing the machine tool part program data using the local editing capabilities of the controller; and setting the feed rate override (FRO), spindle speed override (SSO), or the traverse rate override (TRO)**." (3:16-27). Hutchins discloses updating the user interface – window display – as the part program is running. "Subprogram 100 signals (sends a message to) the display program to **update a window display** of the operating program (processing block 104)." (14:53-55)
if the counter exceeds a threshold,	Hutchins discloses the processor is configured to cause the display of a user interface, if the counter exceeds a threshold. Hutchins discloses resetting the user interface when the batch counter exceeds a threshold (signifying the end of a batch). Hutchins next discloses providing a user interface upon the downloading of a machine tool part program. "**If the program has completed the end of the batch then subprogram 100 signals host computer 20 that a batch has been completed**. Additional parts may still be made, but they will be made under an exception condition. The final end to the production of this batch

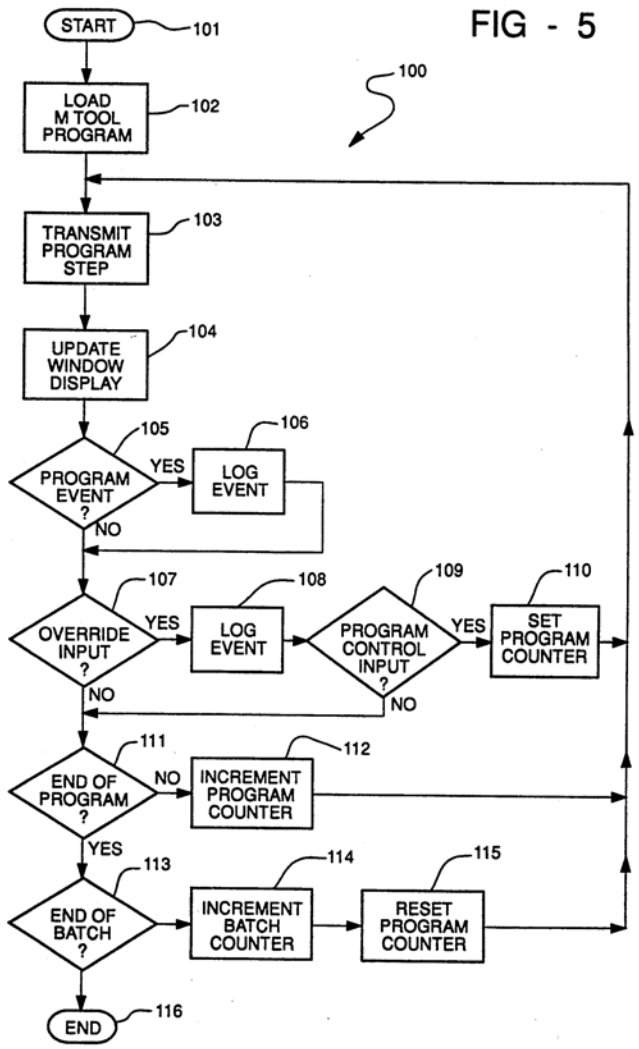
	of parts occurs after the job cleanup **when the machine tool operator downloads a new machine tool part program for the next batch of parts**." (16:25-32) "The execution of the machine tool interface subprogram 200 begins via start block 201. The machine tool is in an idle status until the machine tool operator downloads a machine tool part program. This process involves interaction with local computer 10 via computer user interface 14 and communication with host computer 20. The **machine tool operator must select the desired machine tool part program** from those available at host computer 20**.** This is preferably done via a **menu selection process**. Host computer 20 is aware of the identity of the particular local computer 10 and preferably offers the machine tool operator only those machine tool part programs that are proper for use by the corresponding machine tool(s) 5. Upon selection of a particular machine tool part program, host computer 20 transmits this machine tool part program to local computer 10 via computer network 60. Local computer 10 then stores this machine tool part program within random access memory 16. Running of the particular machine tool part program begins at its first program step." (14:5-25) "The process of running the machine tool part program preferably involves a specification of the batch size. In addition, a batch number for management identification of a particular batch may also be provided. The number of parts in the batch is employed later in control of the machine tool operation. The setting of the batch size and batch I.D. are preferably events signaled to host computer 20. Other data for identification of the job for management reports may also be required. Entry of all this data preferably occurs within processing block 102." (14:26-36) "Once this process is complete, then the machine tool operates according to the selected machine tool part program. This process involves a loop that repeats the steps necessary to machine a part until the machine tool part program is complete. Completion of the batch of parts involves the execution of the machine tool part program enough times to manufacture the number of parts specified by the batch size." (14:37-44) The user interface is displayed upon completion of a program step after a program counter increments and after the program counter is

	exceeded. "Update Window Display 104" (Fig. 5) "Subprogram 100 signals (sends a message to) the display program to **update a window display** of the operating program (processing block 104)." (14:54-56) "Subprogram 100 next tests to determine whether the just transmitted program step is a programmed significant event (decision block 105)." (15:1-3) "The present invention also provides the capability for marking other special events in the machine tool part program. An engineer or master machine tool operator charged with debugging or improving a machine tool part program thus may flag particular portions of the program in order to observe its operation during actual manufactureof parts." (15:12-18) "Whether a program event has been detected, subprogram 100 is signalled (sic) when a machine tool operator initiated event is received (decision block 107)." (15:29-31)
cause the memory to store an input received from the user interface, and	Hutchins discloses that the processor is configured to cause the memory to store an input received from the user interface. Hutchins discloses that memory stores the events recorded by the machine tool operator. "As will be described below, an event is logged by identifying the type of event, date and time of occurrence and **storing this data in an event buffer within random access memory 16**." (13:17-20) "If such a program event is detected, then subprogram 100 logs this event (processing block 106). This process involves writing the identity and the date and time of occurrence of this event to the **reserved event buffer within random access memory 16**." (15:19-23) "Upon detection of any such operator initiated event, the i**dentity and date and time are logged** (processing block 108) in the manner previously described." (15:47-50)
cause the transmitter to transmit the input to a server.	Hutchins discloses that the processor is configured to cause the transmitter to transmit the input to a server. Hutchins discloses transmission of the recorded events stored in random access memory to the host computer ("server"). "As explained above, a host communication program

	running on the local computer 10 is signalled (sic) about the logging of such an event and **transmits this event to host computer 20**." (15:24-28) "Note that the host communications program **transmits any such logged event to host computer 20**." (15:50-52)

Claim 2	Disclosure In Hutchins
The unit of claim 1,	Hutchins discloses the unit of claim 1 as described above.
wherein the input reflects a request to schedule maintenance.	Hutchins discloses that the input from the unit of claim 1 is used for maintenance. "It has heretofore been proposed to integrate the operation of numerically controlled machine tools into a computer network. The typical reason for construction of such machine tool computer networks is the simplified access to and **maintenance of the machine tool part programs** for the numerically controlled machine tools." (1:12-17) It is understood from these teachings of Hutchins that the operator's request is to schedule maintenance.

Claim 5	Disclosure in Hutchins
The unit of claim 1, wherein the processor is further configured to:	Hutchins discloses the unit of claim 1 as described above.
monitor the product for an occurrence in the product of a second trigger event of the predefined plurality of trigger events, and	Hutchins discloses monitoring the product for an occurrence in the product of a second trigger event of the predefined plurality of trigger events, Hutchins discloses monitoring the local computer and machine tool for an occurrence in the product of the completion of a program step ("first trigger event") and the completion of a part program ("second trigger event"). "Subprogram 100 next tests to determine **whether the just transmitted program step** is at the **end of the machine tool part program** (decision block 111). If this is not the end of the part program then the program counter is incremented (processing block 112)." (16:3-7)

	 (Fig. 5)
increment a second counter corresponding to the second trigger event upon detection of the occurrence of the second trigger event in the product.	Hutchins discloses incrementing a second counter corresponding to the second trigger event upon detection of the occurrence of the second trigger event in the product. Hutchins discloses incrementing a batch counter upon completion of a part program ("second trigger event"). "Subprogram 100 then tests to determine whether the end of the batch has been reached. If the batch has not been reached then subprogram 100 i**ncrements the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20)

Claim 6	Disclosure in Hutchins

The unit of claim 5, wherein the processor is further configured to:	Hutchins discloses the unit of claim 5 as described above.
cause the memory to store the second counter; and	Hutchins discloses that the processor is configured to cause the memory to store the second counter; Hutchins discloses a batch counter ("second counter") that is implemented in the local computer. "Subprogram 100 then tests to determine whether the end of the batch has been reached. If the batch has not been reached then subprogram 100 i**ncrements the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20) It is understood from this disclosure that the second counter is stored in memory.
cause the transmitter to transmit a value of the second counter.	Hutchins discloses that the processor is configured to cause the transmitter to transmit a value of the second counter. Hutchins discloses signaling to the host computer that a batch has been completed ("value of the second counter"). "The setting of the **batch size and batch I.D.** are preferably **events signaled to host computer 20**." (14:31-33) "If the program has completed the end of the batch then subprogram 100 **signals host computer 20 that a batch has been completed**." (16:25-27) It is understood from this disclosure that the value of the second counter is transmitted to the host computer.

Claim 7	Disclosure in Hutchins
The unit of claim 1,	Hutchins discloses the unit of claim 1 as described above.
wherein one trigger event of the predefined plurality of trigger events is an exiting of a feature of the product without a use of the feature.	Hutchins discloses one of the predefined plurality of trigger events is an exiting of a feature of the product without a use of the feature. Hutchins discloses exiting a tool status display ("exiting a feature of the product") without changing the tool status display ("without a use of the feature"). "This machine tool status display preferably includes information regarding the current production quantity

	completed and the completed proportion of the manufacture of the present part. Program 200 tests to determine **whether the computer user desires to exit tool status display** (decision block 205). If the computer user **does not wish to change the tool status display then program** 200 delays for a predetermined polling interval (processing block 206) and refreshes the display with the then current information." (17:58-67)

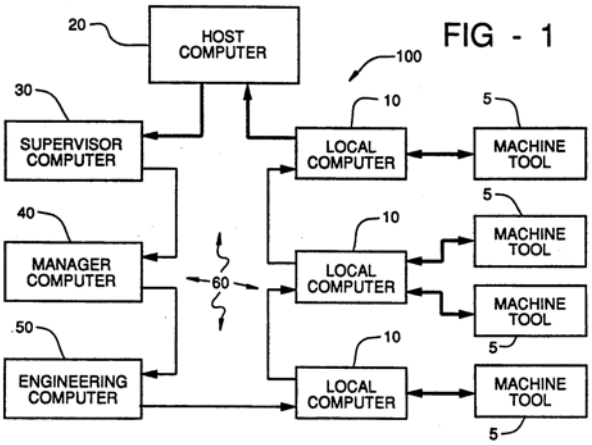
Claim 8	Disclosure In Hutchins
The unit of claim 1,	Hutchins discloses the unit of claim 1 as described above.
wherein one of the predefined plurality of trigger events is a problem associated with the product.	Hutchins discloses that one of the predefined plurality of trigger events is a problem associated with the product. Hutchins discloses the capability to mark a special event in the machine tool part program such as flagging portions of the program for debugging ("problem associated with the product"). "The present invention also provides the capability for **marking other special events** in the machine tool part program. An engineer or master machine tool operator charged with **debugging** or improving a machine tool part program thus **may flag particular portions of the program** in order to observe its operation during actual manufacture of parts." (15:12-18)

Claim 9	Disclosure In Hutchins
The method [sic] of claim 8,	Hutchins discloses the method [sic] of claim 8 as described above.
wherein the problem is an equipment problem.	Hutchins discloses that the problem is an equipment problem. Hutchins discloses the capability to mark a special event in the machine tool part program such as flagging portions of the program for debugging ("problem associated with the product"). "The present invention also provides the capability for marking other special events in the machine tool part program. An engineer or master machine tool operator charged with debugging or improving a machine tool part program thus may flag particular portions of the program in order to observe its operation during actual manufacture of parts." (15:12-18) It is understood the need for debugging could be caused by an equipment problem.

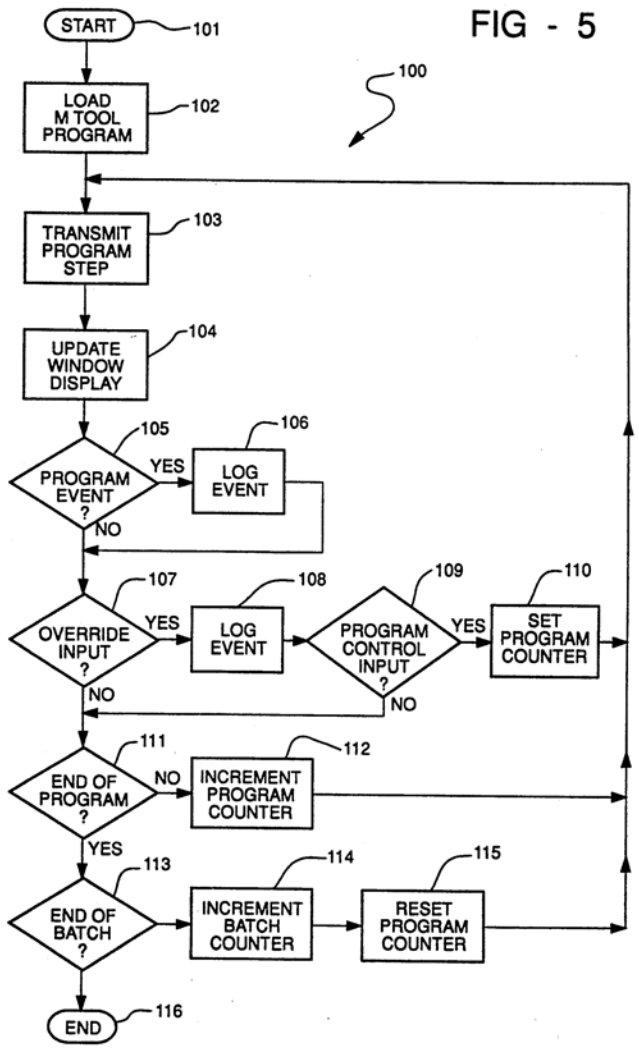
Claim 10	Disclosure In Hutchins
The unit of claim 1,	Hutchins discloses the unit of claim 1 as described above.
wherein a trigger event of the predefined plurality of trigger events is a use of at least one product feature.	Hutchins discloses that a trigger event of the predefined plurality of trigger events is a use of at least one product feature. Hutchins discloses numerous trigger events such as feed rate override, spindle speed override, and traverse rate override commands entered by the machine tool operator. "Lastly, an event is preferably also logged if the machine tool operator exercises the **feed rate override (FRO), spindle speed override (SSO) or the traverse rate override (TRO)**. The operation of any of these overrides will generate a machine tool operator initiated significant event that will be recorded." (15:42-45) It is understood that these commands are a use of at least one product feature.

Claim 14	Disclosure In Hutchins
The unit of claim 1,	Hutchins discloses the unit of claim 1 as described above.
wherein the processor is further configured to increment the counter corresponding to the trigger event upon detection of a second occurrence of the trigger event.	Hutchins discloses that the processor is further configured to increment the counter corresponding to the trigger event upon detection of a second occurrence of the trigger event. Hutchins discloses incrementing a program counter incremented after the completion of each "program step" in a series of program steps. "Subprogram 100 next tests to determine whether the just transmitted program step is the end of the machine tool part program (decision block 111). **If this is not the end of the part program then the program counter is incremented (processing block 112).** Program control then returns to processing block 103 to **repeat the loop by transmission of the next program step (processing block 104)**." (16:3-10) Hutchins discloses incrementing a batch counter upon each completion of an entire part program. "If the last transmitted program step was the **end of the program, then a part has been completed**. This is a program event and would have been detected at decision block 105 and logged at processing block 106." (16:10-14) "Subprogram 100 then tests to determine whether the end

	of the batch has not been reached then subprogram 100 **increments the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115).” (16:15-20)

Claim 15	Disclosure In Hutchins
A method, comprising:	Hutchins a method. “**A method** for controlling one or more numerically controlled machine tools...” (Claim 1: 19:28-29)
monitoring a product for an occurrence in the product of a trigger event of a predefined plurality of trigger events;	Hutchins discloses monitoring a product for an occurrence in the product of a trigger event of a predefined plurality of trigger events. Hutchins discloses a local computer connected to a machine tool (in combination, the “product”).  (Fig. 1) Hutchins discloses a plurality of trigger events including, but not limited to, input from a touch screen display, input from a keyboard, entering a program event, completion of a part program step, and completion of a part program. “Of local computer 10 and the corresponding machine tool(s) 5, and input devices, i.e., **touch screen keyboard, buttons that produce input signals** as software interrupts that permit the machine tool operator to direct the operation of the system.” (6:39-43) “A further subprogram under the run machine tool program checks to determine whether a machine tool part program requires the **logging of an event**. This can occur upon supply of particular machine tool part program steps containing significant events to the machine tool 5 or when the machine tool operator has pressed particular

	control buttons. As will be described below an event is logged by identifying the type of event, date and time of occurrence and storing this data in an event buffer within random access memory 16. The logging of an event signals the host communications subprogram that dispatches a message to host computer 20 via the network 60." (13:11-23) "When the machine tool operator presses the cycle start button on the machine tool, this loop begins by transmitting the next **program step** to the machine tool (processing block 103). In the initial operation of the machine tool part program this **next program step** is the **first program step**. As previously disclosed, this process takes place via machine tool interface 17, which communicates with machine tool 5." (14:45-52) "Once this process is complete, then the machine tool operates according to the selected machine tool part program. This process involves a loop that repeats the steps necessary to machine a part until the machine tool **part program is complete**. Completion of the batch of parts involves the execution of the machine tool part program enough times to manufacture the number of parts specified by the batch size." (14:37-44)
incrementing a counter corresponding to the trigger event upon detection of the occurrence of the trigger event in the product;	Hutchins discloses incrementing a counter corresponding to the trigger event upon detection of the occurrence of the trigger event in the product. Hutchins discloses incrementing a program counter after a program step. "Subprogram 100 next tests to determine whether the just transmitted program step is at the end of the machine tool part program (decision block 111). If this is not the end of the part program then the **program counter is incremented** (processing block 112)." (16:3-7) Hutchins discloses incrementing a counter after the completion of a part program. "Subprogram 100 then tests to determine whether the end of the batch has not been reached then subprogram 100 **increments the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20) Hutchins discloses incrementing the program counter (112) and the batch counter (114) in Figure 5.

	 (Fig. 5)
displaying a user interface, configured to probe for information regarding a use of the product,	Hutchins discloses displaying a user interface, configured to probe for information regarding a use of the product. Hutchins discloses a user interface. "Local computer 10 includes central processing unit 11, network interface 12, real time clock 13, computer user interface 14, memory including read only memory 16 and random access read/write memory 16, and at least one machine tool interface 17." (6:6-10) Hutchins discloses displaying a user interface, configured to probe for information regarding a use of the product. "The **computer user interface 14** permits t**wo way interaction** between the machine tool operator and computer 10. **Computer user interface 14** may include a video display, flat panel display, touch screen, keyboard, or buttons as appropriate and serves as the user

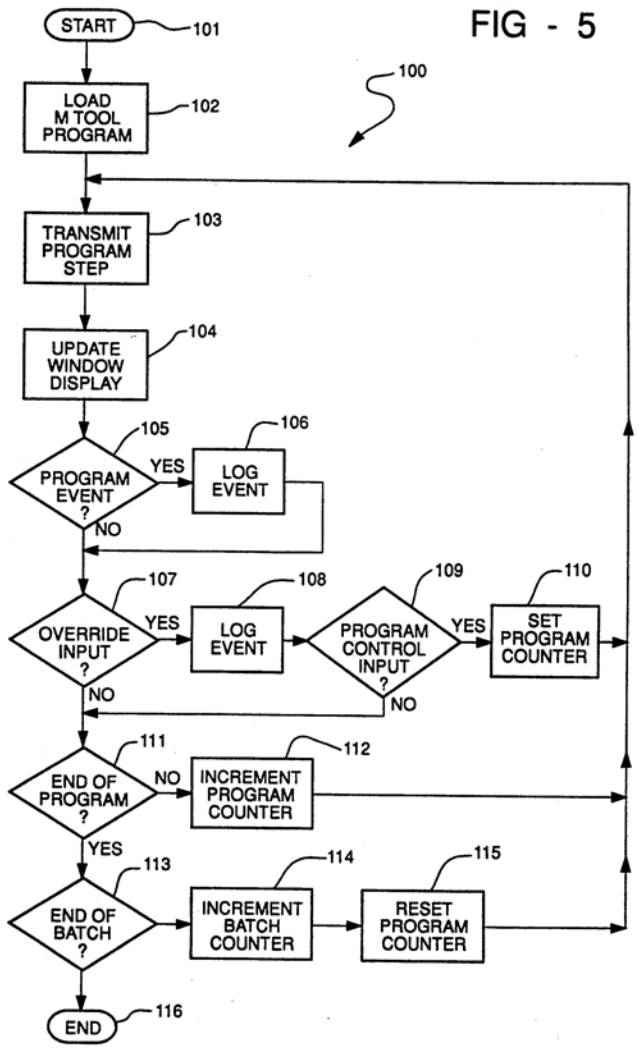
	interface of local computer 10. The **computer user interface 14** includes a display that produces a visual image corresponding to display signals from central processing unit 11 for the user." (6:30-38) "Of local computer 10 and the corresponding machine tool(s) 5, and input devices, i.e., touch screen keyboard, buttons that produce **input signals** as software interrupts that permit the machine tool operator to **direct the operation of the system**." (6:39-43) "The other type of events reported include operator initiated events at the machine tool. These machine tool operator initiated events include, but are not limited to: **down-loading a machine tool part program to the machine tool; setting the batch size; beginning or ending the operation cycle of the machine tool part program; skipping or deleting operations such as may occur when reworking a work piece; editing the machine tool part program data using the local editing capabilities of the controller; and setting the feed rate override (FRO), spindle speed override (SSO), or the traverse rate override (TRO)**." (3:16-27). Hutchins discloses updating the user interface – window display – as the part program is running. "Subprogram 100 signals (sends a message to) the display program to **update a window display** of the operating program (processing block 104)." (14:53-55)
if the counter exceeds a threshold;	Hutchins discloses displaying the user interface, if the counter exceeds a threshold. Hutchins discloses resetting the user interface when the batch counter exceeds a threshold (signifying the end of a batch). Hutchins next discloses providing a user interface upon the downloading of a machine tool part program. "If the program has **completed the end of the batch** then subprogram 100 signals host computer 20 that a batch has been completed. Additional parts may still be made, but they will be made under an exception condition. The final end to the production of this batch of parts occurs after the job cleanup **when the machine tool operator downloads a new machine tool part program for the next batch of parts**." (16:25-32) "The execution of the machine tool interface subprogram 200 begins via start block 201. The machine tool is in an

	idle status until the machine tool operator downloads a machine tool part program. This process involves interaction with local computer 10 via computer user interface 14 and communication with host computer 20. The **machine tool operator must select the desired machine tool part program** from those available at host computer 20**.** This is preferably done via a **menu selection process**. Host computer 20 is aware of the identity of the particular local computer 10 and preferably offers the machine tool operator only those machine tool part programs that are proper for use by the corresponding machine tool(s) 5. Upon selection of a particular machine tool part program, host computer 20 transmits this machine tool part program to local computer 10 via computer network 60. Local computer 10 then stores this machine tool part program within random access memory 16. Running of the particular machine tool part program begins at its first program step." (14:5-25) "The process of running the machine tool part program preferably involves a specification of the batch size. In addition, a batch number for management identification of a particular batch may also be provided. The number of parts in the batch is employed later in control of the machine tool operation. The setting of the batch size and batch I.D. are preferably events signaled to host computer 20. Other data for identification of the job for management reports may also be required. Entry of all this data preferably occurs within processing block 102." (14:26-36) "Once this process is complete, then the machine tool operates according to the selected machine tool part program. This process involves a loop that repeats the steps necessary to machine a part until the machine tool part program is complete. Completion of the batch of parts involves the execution of the machine tool part program enough times to manufacture the number of parts specified by the batch size." (14:37-44) The user interface is displayed upon completion of a program step after a program counter increments and after the program counter is exceeded. See, Fig. 5. ("Update Window Display 104") "Subprogram 100 signals (sends a message to) the display program to **update a window display** of the

	operating program (processing block 104).” (14:53-55) “Subprogram 100 next tests to determine whether the just transmitted program step is a programmed significant event (decision block 105).” (15:1-3) “The present invention also provides the capability for marking other special events in the machine tool part program. An engineer or master machine tool operator charged with debugging or improving a machine tool part program thus may flag particular portions of the program in order to observe its operation during actual manufactureof parts.” (15:12-18) “Whether a program event has been detected, subprogram 100 is signalled (sic) when a machine tool operator initiated event is received (decision block 107).” (15:29-31)
storing an input received from the user interface on a device; and	Hutchins discloses storing an input received from the user interface on a device. Hutchins discloses that memory stores the events recorded by the machine tool operator. “As will be described below, an event is logged by identifying the type of event, date and time of occurrence and **storing this data in an event buffer within random access memory 16**.” (13:17-20) “If such a program event is detected, then subprogram 100 logs this event (processing block 106). This process involves writing the identity and the date and time of occurrence of this event to the **reserved event buffer within random access memory 16**.” (15:19-23) “Upon detection of any such operator initiated event, the i**dentity and date and time are logged** (processing block 108) in the manner previously described.” (15:47-50)
transmitting the input to a server.	Hutchins discloses transmitting the input to a server. Hutchins discloses transmission of the recorded events stored in random access memory to the host computer (“server”). “As explained above, a host communication program running on the local computer 10 is signalled (sic) about the logging of such an event and **transmits this event to host computer 20.”** (15:24-28) “Note that the host communications program **transmits any such logged event to host computer 20**.” (15:50-

	52)

Claim 17	Disclosure In Hutchins
The method of claim 15, further comprising:	Hutchins discloses the method of claim 15 as described above.
monitoring the product for an occurrence in the product of a second trigger event of the predefined plurality of trigger events;	Hutchins discloses monitoring the product for an occurrence in the product of a second trigger event of the predefined plurality of trigger events. Hutchins discloses monitoring the local computer and machine tool for an occurrence in the product of the completion of a program step ("first trigger event") and the completion of a part program ("second trigger event"). "Subprogram 100 next tests to determine **whether the just transmitted program step** is at the **end of the machine tool part program** (decision block 111). If this is not the end of the part program then the program counter is incremented (processing block 112)." (16:3-7)

	 (Fig. 5)
incrementing a value of a second counter corresponding to the second trigger event upon detection of the second trigger event in the product.	Hutchins discloses incrementing a value of a second counter corresponding to the second trigger event upon detection of the second trigger event in the product. Hutchins discloses incrementing a batch counter upon completion of a part program ("second trigger event"). "Subprogram 100 then tests to determine whether the end of the batch has been reached. If the batch has not been reached then subprogram 100 i**ncrements the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20)

Claim 18	Disclosure In Hutchins

The method of claim 15, further comprising:	Hutchins discloses the method of claim 15 as described above.
storing the second counter on the device; and	Hutchins discloses storing the second counter on the device. Hutchins discloses a batch counter ("second counter") that is implemented in the local computer. "Subprogram 100 then tests to determine whether the end of the batch has been reached. If the batch has not been reached then subprogram 100 i**ncrements the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20) It is understood from this disclosure that the second counter is stored in memory.
transmitting a value of the second counter to the server.	Hutchins discloses signaling to the host computer that a batch has been completed ("value of the second counter"). "The setting of the **batch size and batch I.D.** are preferably **events signaled to host computer 20**." (14:31-33) "If the program has completed the end of the batch then subprogram 100 **signals host computer 20 that a batch has been completed.**" (16:25-27) It is understood from this disclosure that the value of the second counter is transmitted to the host computer.

Claim 19	Disclosure In Hutchins
The method of claim 15,	Hutchins discloses the method of claim 15 as described above.
wherein one of the predefined plurality of trigger events is a problem associated with the product.	Hutchins discloses that one of the predefined plurality of trigger events is a problem associated with the product. Hutchins discloses the capability to mark a special event in the machine tool part program such as flagging portions of the program for debugging ("problem associated with the product"). "The present invention also provides the capability for **marking other special events** in the machine tool part program. An engineer or master machine tool operator charged with **debugging** or improving a machine tool part program thus **may flag particular portions of the program** in order to observe its operation during actual manufacture of parts." (15:12-18)

Claim 20	Disclosure In Hutchins
The method of claim 15,	Hutchins discloses the method of claim 15 as described above.
wherein one of the predefined plurality of trigger events is an exiting of a feature of the product without a use of the feature.	Hutchins discloses one of the predefined plurality of trigger events is an exiting of a feature of the product without a use of the feature. Hutchins discloses exiting a tool status display ("exiting a feature of the product") without changing the tool status display ("without a use of the feature"). "This machine tool status display preferably includes information regarding the current production quantity completed and the completed proportion of the manufacture of the present part. Program 200 tests to determine **whether the computer user desires to exit tool status display** (decision block 205). If the computer user **does not wish to change the tool status display then program** 200 delays for a predetermined polling interval (processing block 206) and refreshes the display with the then current information." (17:58-67)

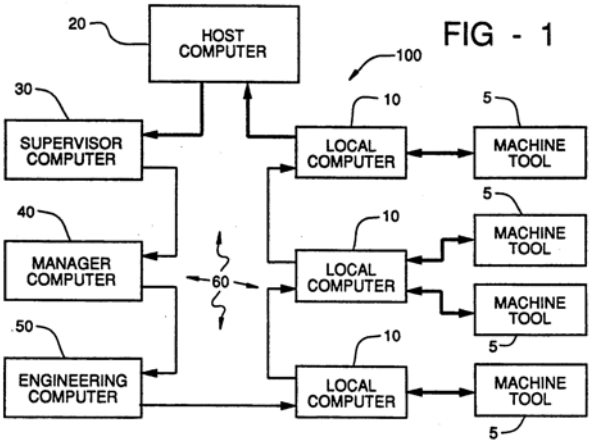
Claim 21	Disclosure In Hutchins
The method of claim 19,	Hutchins discloses the method of claim 19 as described above.
wherein the problem is an equipment problem.	Hutchins discloses that the problem is an equipment problem. Hutchins discloses the capability to mark a special event in the machine tool part program such as flagging portions of the program for debugging ("problem associated with the product"). "The present invention also provides the capability for marking other special events in the machine tool part program. An engineer or master machine tool operator charged with debugging or improving a machine tool part program thus may flag particular portions of the program in order to observe its operation during actual manufacture of parts." (15:12-18) It is understood the need for debugging could be caused by an equipment problem.

Claim 22	Disclosure In Hutchins
The method of claim 15,	Hutchins discloses the method of claim 15 as described above.

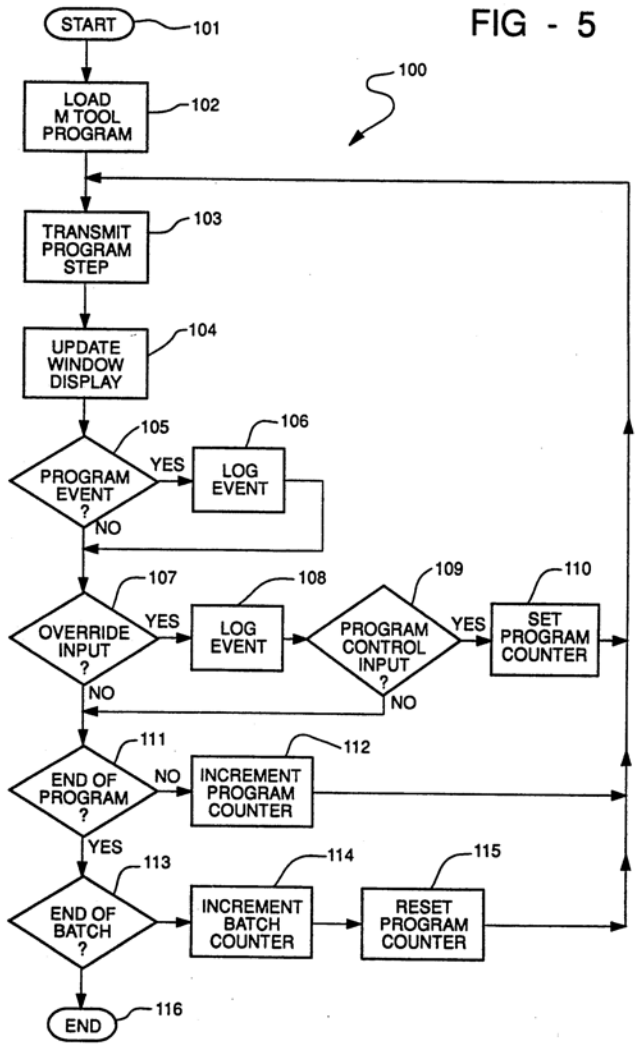
wherein one of the predefined plurality of trigger events is a use of at least one product feature.	Hutchins discloses that one of the predefined plurality of trigger events is a use of at least one product feature. Hutchins discloses numerous trigger events such as feed rate override, spindle speed override, and traverse rate override commands entered by the machine tool operator. "Lastly, an event is preferably also logged if the machine tool operator exercises the **feed rate override (FRO), spindle speed override (SSO) or the traverse rate override (TRO)**. The operation of any of these overrides will generate a machine tool operator initiated significant event that will be recorded." (15:42-45) It is understood that these commands are a use of at least one product feature.

Claim 26	Disclosure In Hutchins
The method of claim 15,	Hutchins discloses the method of claim 15 as described above.
further comprising: incrementing the counter corresponding to the trigger event upon detection of a second occurrence of the trigger event in the product.	Hutchins discloses incrementing the counter corresponding to the trigger event upon detection of a second occurrence of the trigger event in the product. Hutchins incrementing a program counter upon completion of each part program step "Subprogram 100 next tests to determine whether the just transmitted program step is the end of the machine tool part program (decision block 111). If this is not the end of the part program then the program counter is incremented (processing block 112). Program control then returns to processing block 103 to **repeat the loop by transmission of the next program step (processing block 104)**. If the last transmitted program step was the end of the program, then a part has been completed. This is a program event and would have been detected at decision block 105 and logged at processing block 106." (16:3-14)

Claim 27	Disclosure In Hutchins
A tangible computer readable medium having stored thereon, computer executable instructions that, if	Hutchins discloses a tangible computer readable medium having stored thereon, computer executable instructions that, if executed by a computing device, cause the computing device to perform a method. Hutchins discloses a computer including a CPU and memory containing instructions to perform the method of claim 27.

executed by a computing device, cause the computing device to perform a method comprising:	"Each local computer 10 includes a MAIN program which orchestrates the coordination of subsidiary programs serving as the user interface between local computer 10 and the computer user interface 14, the machine tool interface 17 and the corresponding machine tool 5." (12:44-49) "One of the subprograms available to the user of local computer 10 runs a machine tool 5. This run machine tool subprogram reads each block of the machine tool part program data and energizes the machine tool 5 and its servo systems through the machine tool interface 17 according to the instructions contained in the block. This subprogram will be described in further detail in conjunction with FIG. 5." (13:3-10).
monitoring a product for an occurrence in the product of a trigger event of a predefined plurality of trigger events,	Hutchins discloses monitoring a product for an occurrence in the product of a trigger event of a predefined plurality of trigger events. Hutchins discloses a local computer connected to a machine tool (in combination, the "product").  (Fig. 1) Hutchins discloses a plurality of trigger events including, but not limited to, input from a touch screen display, input from a keyboard, entering a program event, completion of a part program step, and completion of a part program. "Of local computer 10 and the corresponding machine tool(s) 5, and input devices, i.e., **touch screen keyboard, buttons that produce input signals** as software interrupts that permit the machine tool operator to direct the operation of the system." (6:39-43) "A further subprogram under the run machine tool

	program checks to determine whether a machine tool part program requires the **logging of an event**. This can occur upon supply of particular machine tool part program steps containing significant events to the machine tool 5 or when the machine tool operator has pressed particular control buttons. As will be described below an event is logged by identifying the type of event, date and time of occurrence and storing this data in an event buffer within random access memory 16. The logging of an event signals the host communications subprogram that dispatches a message to host computer 20 via the network 60." (13:11-23) "When the machine tool operator presses the cycle start button on the machine tool, this loop begins by transmitting the next **program step** to the machine tool (processing block 103). In the initial operation of the machine tool part program this **next program step** is the **first program step**. As previously disclosed, this process takes place via machine tool interface 17, which communicates with machine tool 5." (14:45-52) "Once this process is complete, then the machine tool operates according to the selected machine tool part program. This process involves a loop that repeats the steps necessary to machine a part until the machine tool **part program is complete**. Completion of the batch of parts involves the execution of the machine tool part program enough times to manufacture the number of parts specified by the batch size." (14:37-44)
incrementing a counter corresponding to the trigger event upon detection of the occurrence of the trigger event in the product;	Hutchins discloses incrementing a counter corresponding to the trigger event upon detection of the occurrence of the trigger event in the product. Hutchins discloses incrementing a program counter after a program step. "Subprogram 100 next tests to determine whether the just transmitted program step is at the end of the machine tool part program (decision block 111). If this is not the end of the part program then the **program counter is incremented** (processing block 112)." (16:3-7) Hutchins discloses incrementing a counter after the completion of a part program. "Subprogram 100 then tests to determine whether the end of the batch has not been reached then subprogram 100 **increments the batch counter** (processing block 114) and resets the program counter to the beginning of the

	program (processing block 115).” (16:15-20) Hutchins discloses incrementing the program counter (112) and the batch counter (114) in Figure 5.  (Fig. 5)
displaying a user interface, configured to probe for information regarding a use of the product,	Hutchins discloses displaying a user interface, configured to probe for information regarding a use of the product. Hutchins discloses a user interface. “Local computer 10 includes central processing unit 11, network interface 12, real time clock 13, computer user interface 14, memory including read only memory 16 and random access read/write memory 16, and at least one machine tool interface 17.” (6:6-10) Hutchins discloses displaying a user interface, configured to probe for information regarding a use of the product.

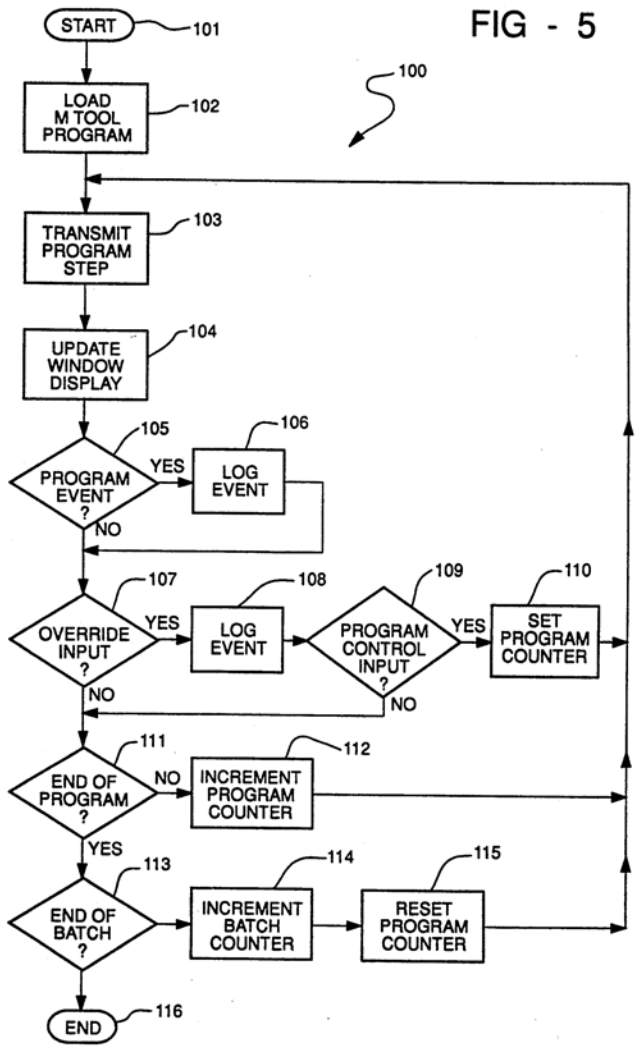
	"The **computer user interface 14** permits t**wo way interaction** between the machine tool operator and computer 10. **Computer user interface 14** may include a video display, flat panel display, touch screen, keyboard, or buttons as appropriate and serves as the user interface of local computer 10. The **computer user interface 14** includes a display that produces a visual image corresponding to display signals from central processing unit 11 for the user." (6:30-38) "Of local computer 10 and the corresponding machine tool(s) 5, and input devices, i.e., touch screen keyboard, buttons that produce **input signals** as software interrupts that permit the machine tool operator to **direct the operation of the system**." (6:39-43) "The other type of events reported include operator initiated events at the machine tool. These machine tool operator initiated events include, but are not limited to: **down-loading a machine tool part program to the machine tool; setting the batch size; beginning or ending the operation cycle of the machine tool part program; skipping or deleting operations such as may occur when reworking a work piece; editing the machine tool part program data using the local editing capabilities of the controller; and setting the feed rate override (FRO), spindle speed override (SSO), or the traverse rate override (TRO).**" (3:16-27). Hutchins discloses updating the user interface – window display – as the part program is running. "Subprogram 100 signals (sends a message to) the display program to **update a window display** of the operating program (processing block 104)." (14:53-55)
if the counter exceeds a threshold;	Hutchins discloses displaying the user interface, if the counter exceeds a threshold. Hutchins discloses resetting the user interface when the batch counter exceeds a threshold (signifying the end of a batch). Hutchins next discloses providing a user interface upon the downloading of a machine tool part program. "If the program has **completed the end of the batch** then subprogram 100 signals host computer 20 that a batch has been completed. Additional parts may still be made, but they will be made under an exception condition. The final end to the production of this batch of parts occurs

	after the job cleanup **when the machine tool operator downloads a new machine tool part program for the next batch of parts**." (16:25-32) "The execution of the machine tool interface subprogram 200 begins via start block 201. The machine tool is in an idle status until the machine tool operator downloads a machine tool part program. This process involves interaction with local computer 10 via computer user interface 14 and communication with host computer 20. The **machine tool operator must select the desired machine tool part program** from those available at host computer 20**.** This is preferably done via a **menu selection process**. Host computer 20 is aware of the identity of the particular local computer 10 and preferably offers the machine tool operator only those machine tool part programs that are proper for use by the corresponding machine tool(s) 5. Upon selection of a particular machine tool part program, host computer 20 transmits this machine tool part program to local computer 10 via computer network 60. Local computer 10 then stores this machine tool part program within random access memory 16. Running of the particular machine tool part program begins at its first program step." (14:5-25) "The process of running the machine tool part program preferably involves a specification of the batch size. In addition, a batch number for management identification of a particular batch may also be provided. The number of parts in the batch is employed later in control of the machine tool operation. The setting of the batch size and batch I.D. are preferably events signaled to host computer 20. Other data for identification of the job for management reports may also be required. Entry of all this data preferably occurs within processing block 102." (14:26-36) "Once this process is complete, then the machine tool operates according to the selected machine tool part program. This process involves a loop that repeats the steps necessary to machine a part until the machine tool part program is complete. Completion of the batch of parts involves the execution of the machine tool part program enough times to manufacture the number of parts specified by the batch size." (14:37-44) The user interface is displayed upon completion of a program step after a program counter increments and after the program counter is

	exceeded. "Update Window Display 104" (Fig. 5) "Subprogram 100 signals (sends a message to) the display program to **update a window display** of the operating program (processing block 104)." (14:53-55) "Subprogram 100 next tests to determine whether the just transmitted program step is a programmed significant event (decision block 105)." (15:1-3) "The present invention also provides the capability for marking other special events in the machine tool part program. An engineer or master machine tool operator charged with debugging or improving a machine tool part program thus may flag particular portions of the program in order to observe its operation during actual manufactureof parts." (15:12-18) "Whether a program event has been detected, subprogram 100 is signalled (sic) when a machine tool operator initiated event is received (decision block 107)." (15:29-31)
storing an input received from the user interface on a device; and	Hutchins discloses that memory stores the events recorded by the machine tool operator. Hutchins discloses that memory stores the events recorded by the machine tool operator. "As will be described below, an event is logged by identifying the type of event, date and time of occurrence and **storing this data in an event buffer within random access memory 16**." (13:17-20) "If such a program event is detected, then subprogram 100 logs this event (processing block 106). This process involves writing the identity and the date and time of occurrence of this event to the **reserved event buffer within random access memory 16**." (15:19-23) "Upon detection of any such operator initiated event, the i**dentity and date and time are logged** (processing block 108) in the manner previously described." (15:47-50)
transmitting the input to a server.	Hutchins discloses transmitting the input to a server. Hutchins discloses transmission of the recorded events stored in random access memory to the host computer ("server"). "As explained above, a host communication program running on the local computer 10 is signalled (sic) about

	the logging of such an event and **transmits this event to host computer 20**." (15:24-28) "Note that the host communications program **transmits any such logged event to host computer 20**." (15:50-52)

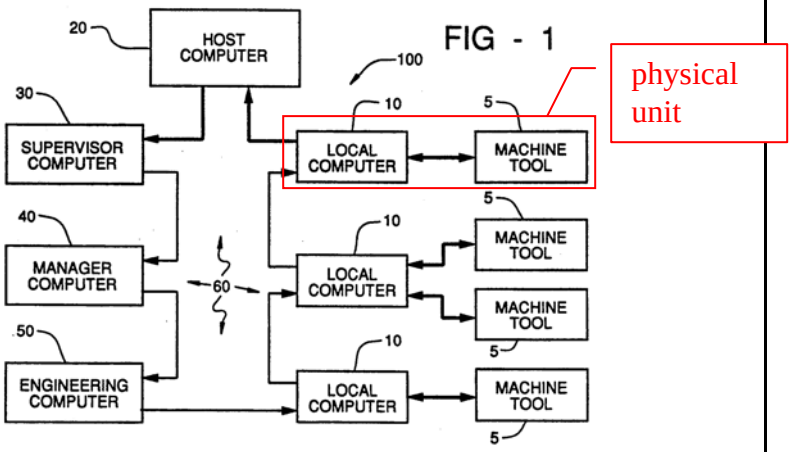
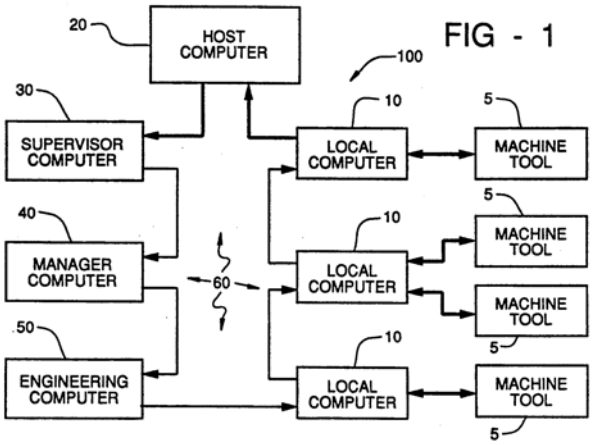
Claim 28	Disclosure In Hutchins
The tangible computer readable medium of claim 27, wherein the monitoring further includes:	Hutchins discloses the method of claim 27 as described above.
monitoring the product for an occurrence in the product of a second trigger event of the predefined plurality of trigger events; and	Hutchins discloses monitoring the product for an occurrence in the product of a second trigger event of the predefined plurality of trigger events. Hutchins discloses monitoring the local computer and machine tool for an occurrence in the product of the completion of a program step ("first trigger event") and the completion of a part program ("second trigger event"). "Subprogram 100 next tests to determine **whether the just transmitted program step** is at the **end of the machine tool part program** (decision block 111). If this is not the end of the part program then the program counter is incremented (processing block 112)." (16:3-7)

	 (Fig. 5)
incrementing a value of a second counter corresponding to the second trigger event upon detection of the occurrence of the second trigger event in the product.	Hutchins discloses incrementing a value of a second counter corresponding to the second trigger event upon detection of the second trigger event in the product. Hutchins discloses incrementing a batch counter upon completion of a part program ("second trigger event"). "Subprogram 100 then tests to determine whether the end of the batch has been reached. If the batch has not been reached then subprogram 100 i**ncrements the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20)

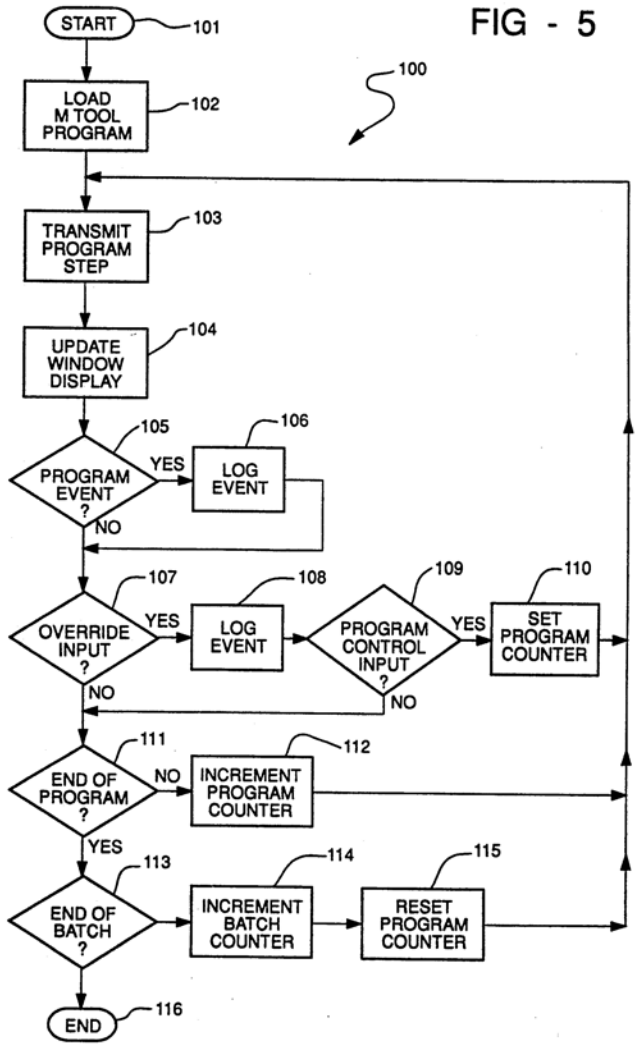
Claim 29	Disclosure In Hutchins

The tangible computer readable medium of claim 27 [sic], wherein the method further includes:	Hutchins discloses the method of claim 27 [sic] as described above.
storing the second counter on the device; and	Hutchins discloses storing the second counter on the device. Hutchins discloses a batch counter ("second counter") that is implemented in the local computer. "Subprogram 100 then tests to determine whether the end of the batch has been reached. If the batch has not been reached then subprogram 100 i**ncrements the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20) It is understood from this disclosure that the second counter is stored in memory.
transmitting the value of the second counter to the server.	Hutchins discloses transmitting a value of the second counter to the server. Hutchins discloses signaling to the host computer that a batch has been completed ("value of the second counter"). "The setting of the **batch size and batch I.D.** are preferably **events signaled to host computer 20**." (14:31-33) "If the program has completed the end of the batch then subprogram 100 **signals host computer 20 that a batch has been completed**." (16:25-27) It is understood from this disclosure that the value of the second counter is transmitted to the host computer.

Claim 30	Disclosure In Hutchins
A physical unit, comprising:	Hutchins discloses a physical unit. Hutchins discloses a local computer (10) connected to a machine tool (5) (collectively, "physical unit").

	 (Fig. 1) Hutchins also discloses a computer numerically controlled machine ("physical unit"). "This technique may be used with current **computer numerically controlled machines** by revising their control software-firmware to generate the event records and by transmitting same via the serial port to a host computer programmed to accept and store the records." (3:28-32).
means for monitoring a product for an occurrence in the product of a trigger event of a predefined plurality of trigger events;	Hutchins discloses means for monitoring a product for an occurrence in the product of a trigger event of a predefined plurality of trigger events. Hutchins discloses a local computer connected to a machine tool (in combination, the "product").  (Fig. 1) Hutchins discloses a plurality of trigger events including, but not limited to, input from a touch screen display, input from a keyboard, entering a program event, completion of a part program step, and completion of a

	part program. "Of local computer 10 and the corresponding machine tool(s) 5, and input devices, i.e., **touch screen keyboard, buttons that produce input signals** as software interrupts that permit the machine tool operator to direct the operation of the system." (6:39-43) "A further subprogram under the run machine tool program checks to determine whether a machine tool part program requires the **logging of an event**. This can occur upon supply of particular machine tool part program steps containing significant events to the machine tool 5 or when the machine tool operator has pressed particular control buttons. As will be described below an event is logged by identifying the type of event, date and time of occurrence and storing this data in an event buffer within random access memory 16. The logging of an event signals the host communications subprogram that dispatches a message to host computer 20 via the network 60." (13:11-23) "When the machine tool operator presses the cycle start button on the machine tool, this loop begins by transmitting the next **program step** to the machine tool (processing block 103). In the initial operation of the machine tool part program this **next program step** is the **first program step**. As previously disclosed, this process takes place via machine tool interface 17, which communicates with machine tool 5." (14:45-52) "Once this process is complete, then the machine tool operates according to the selected machine tool part program. This process involves a loop that repeats the steps necessary to machine a part until the machine tool **part program is complete**. Completion of the batch of parts involves the execution of the machine tool part program enough times to manufacture the number of parts specified by the batch size." (14:37-44)
means for incrementing a counter corresponding to the trigger event upon detection of the occurrence of the trigger event;	Hutchins discloses means for incrementing a counter corresponding to the trigger event upon detection of the occurrence of the trigger event. Hutchins discloses incrementing a program counter after a program step. "Subprogram 100 next tests to determine whether the just transmitted program step is at the end of the machine tool part program (decision block 111). If this is not the end of the part program then the **program counter is**

	incremented (processing block 112)." (16:3-7) Hutchins discloses incrementing a counter after the completion of a part program. "Subprogram 100 then tests to determine whether the end of the batch has not been reached then subprogram 100 **increments the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20) Hutchins discloses incrementing the program counter (112) and the batch counter (114) in Figure 5.  (Fig. 5)
means for probing for information regarding a use of the product	Hutchins discloses means for probing for information regarding a use of the product if the counter exceeds a threshold.

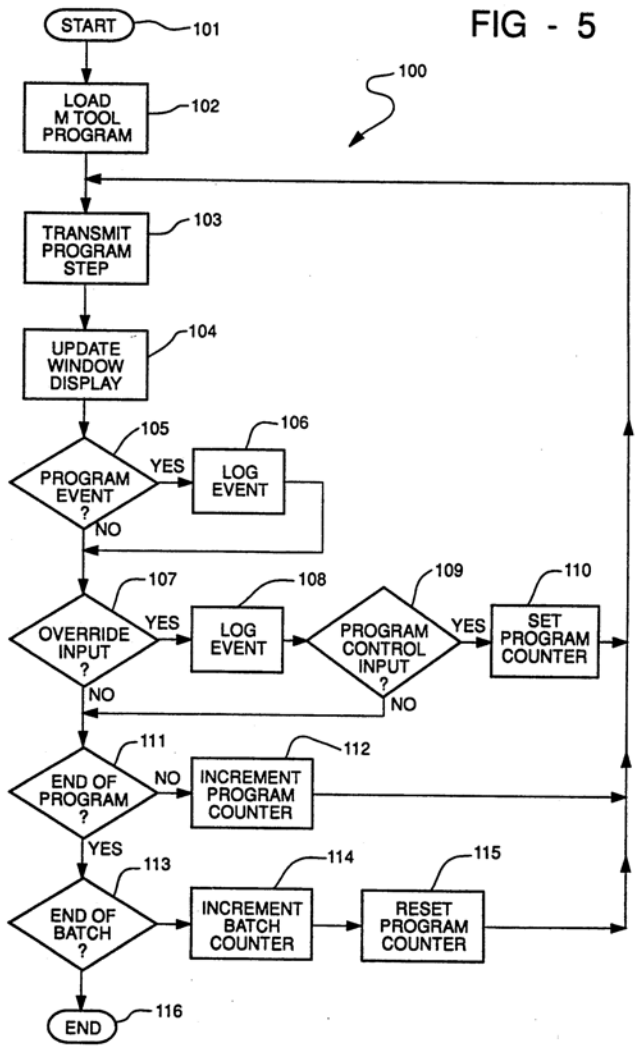
	Hutchins discloses a user interface. "Local computer 10 includes central processing unit 11, network interface 12, real time clock 13, computer user interface 14, memory including read only memory 16 and random access read/write memory 16, and at least one machine tool interface 17." (6:6-10) Hutchins discloses displaying a user interface, configured to probe for information regarding a use of the product. "The **computer user interface 14** permits t**wo way interaction** between the machine tool operator and computer 10. **Computer user interface 14** may include a video display, flat panel display, touch screen, keyboard, or buttons as appropriate and serves as the user interface of local computer 10. The **computer user interface 14** includes a display that produces a visual image corresponding to display signals from central processing unit 11 for the user." (6:30-38) "Of local computer 10 and the corresponding machine tool(s) 5, and input devices, i.e., touch screen keyboard, buttons that produce **input signals** as software interrupts that permit the machine tool operator to **direct the operation of the system**." (6:39-43) "The other type of events reported include operator initiated events at the machine tool. These machine tool operator initiated events include, but are not limited to: **down-loading a machine tool part program to the machine tool; setting the batch size; beginning or ending the operation cycle of the machine tool part program; skipping or deleting operations such as may occur when reworking a work piece; editing the machine tool part program data using the local editing capabilities of the controller; and setting the feed rate override (FRO), spindle speed override (SSO), or the traverse rate override (TRO)**." (3:16-27). Hutchins discloses updating the user interface – window display – as the part program is running. "Subprogram 100 signals (sends a message to) the display program to **update a window display** of the operating program (processing block 104)." (14:53-55)
if the counter exceeds a threshold;	Hutchins discloses probing for information regarding a use of the product if the counter exceeds a threshold.

	Hutchins discloses resetting the user interface when the batch counter exceeds a threshold (signifying the end of a batch). Hutchins next discloses providing a user interface upon the downloading of a machine tool part program. "If the program has **completed the end of the batch** then subprogram 100 signals host computer 20 that a batch has been completed. Additional parts may still be made, but they will be made under an exception condition. The final end to the production of this batch of parts occurs after the job cleanup **when the machine tool operator downloads a new machine tool part program for the next batch of parts**." (16:25-32) "The execution of the machine tool interface subprogram 200 begins via start block 201. The machine tool is in an idle status until the machine tool operator downloads a machine tool part program. This process involves interaction with local computer 10 via computer user interface 14 and communication with host computer 20. The **machine tool operator must select the desired machine tool part program** from those available at host computer 20**.** This is preferably done via a **menu selection process**. Host computer 20 is aware of the identity of the particular local computer 10 and preferably offers the machine tool operator only those machine tool part programs that are proper for use by the corresponding machine tool(s) 5. Upon selection of a particular machine tool part program, host computer 20 transmits this machine tool part program to local computer 10 via computer network 60. Local computer 10 then stores this machine tool part program within random access memory 16. Running of the particular machine tool part program begins at its first program step." (14:5-25) "The process of running the machine tool part program preferably involves a specification of the batch size. In addition, a batch number for management identification of a particular batch may also be provided. The number of parts in the batch is employed later in control of the machine tool operation. The setting of the batch size and batch I.D. are preferably events signaled to host computer 20. Other data for identification of the job for management reports may also be required. Entry of all this data preferably occurs within processing block 102." (14:26-36) "Once this process is complete, then the machine tool operates according to the selected machine tool part

	program. This process involves a loop that repeats the steps necessary to machine a part until the machine tool part program is complete. Completion of the batch of parts involves the execution of the machine tool part program enough times to manufacture the number of parts specified by the batch size." (14:37-44) The user interface is displayed upon completion of a program step after a program counter increments and after the program counter is exceeded. "Update Window Display 104" (Fig. 5) "Subprogram 100 signals (sends a message to) the display program to **update a window display** of the operating program (processing block 104)." (14:54-56) "Subprogram 100 next tests to determine whether the just transmitted program step is a programmed significant event (decision block 105)." (15:1-3) "The present invention also provides the capability for marking other special events in the machine tool part program. An engineer or master machine tool operator charged with debugging or improving a machine tool part program thus may flag particular portions of the program in order to observe its operation during actual manufactureof parts." (15:12-18) "Whether a program event has been detected, subprogram 100 is signalled (sic) when a machine tool operator initiated event is received (decision block 107)." (15:29-31)
means for storing an input received from the means for probing; and	Hutchins discloses means for storing an input received from the means for probing. Hutchins discloses that memory stores the events recorded by the machine tool operator. "As will be described below, an event is logged by identifying the type of event, date and time of occurrence and **storing this data in an event buffer within random access memory 16**." (13:17-20) "If such a program event is detected, then subprogram 100 logs this event (processing block 106). This process involves writing the identity and the date and time of occurrence of this event to the **reserved event buffer within random access memory 16**." (15:19-23) "Upon detection of any such operator initiated event, the i**dentity and date and time are logged** (processing

	block 108) in the manner previously described." (15:47-50)
means for transmitting the input to a server.	Hutchins discloses means for transmitting the input to a server. Hutchins discloses transmission of the recorded events stored in random access memory to the host computer ("server"). "As explained above, a host communication program running on the local computer 10 is signalled (sic) about the logging of such an event and **transmits this event to host computer 20**." (15:24-28) "Note that the host communications program **transmits any such logged event to host computer 20**." (15:50-52)

Claim 31	Disclosure In Hutchins
The unit of claim 30, further comprising:	Hutchins discloses the physical unit of claim 30 as described above.
means for monitoring the product for an occurrence in the product of a second trigger event of the predefined plurality of trigger events; and	Hutchins discloses means for monitoring the product for an occurrence in the product of a second trigger event of the predefined plurality of trigger events. Hutchins discloses monitoring the local computer and machine tool for an occurrence in the product of the completion of a program step ("first trigger event") and the completion of a part program ("second trigger event"). "Subprogram 100 next tests to determine **whether the just transmitted program step** is at the **end of the machine tool part program** (decision block 111). If this is not the end of the part program then the program counter is incremented (processing block 112)." (16:3-7)

	 (Fig. 5)
means for incrementing a value of a second counter corresponding to the second trigger event upon detection of the occurrence of the second trigger event in the product.	Hutchins discloses means for incrementing a value of a second counter corresponding to the second trigger event upon detection of the second trigger event in the product. Hutchins discloses incrementing a batch counter upon completion of a part program ("second trigger event"). "Subprogram 100 then tests to determine whether the end of the batch has been reached. If the batch has not been reached then subprogram 100 i**ncrements the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20)

Claim 32	Disclosure In Hutchins

The unit of claim 30 [sic], further comprising:	Hutchins discloses the unit of claim 30 [sic]as described above.
means for storing the second counter on the device; and	Hutchins discloses means for storing the second counter on the device. Hutchins discloses a batch counter ("second counter") that is implemented in the local computer. "Subprogram 100 then tests to determine whether the end of the batch has been reached. If the batch has not been reached then subprogram 100 i**ncrements the batch counter** (processing block 114) and resets the program counter to the beginning of the program (processing block 115)." (16:15-20) It is understood from this disclosure that the second counter is stored in memory.
means for transmitting the value of the second counter to the server.	Hutchins discloses means for transmitting a value of the second counter to the server. Hutchins discloses signaling to the host computer that a batch has been completed ("value of the second counter"). "The setting of the **batch size and batch I.D.** are preferably **events signaled to host computer 20**." (14:31-33) "If the program has completed the end of the batch then subprogram 100 **signals host computer 20 that a batch has been completed**." (16:25-27) It is understood from this disclosure that the value of the second counter is transmitted to the host computer.