Robot Controllers and Content Files Therefor

The invention relates to control units and methods for controlling a scene of operation of a robot. The invention also relates to an apparatus and method for creating a content file for controlling a scene of operation of a robot. The invention also relates to a content file for controlling a scene of operation of a robot.

The invention has particular, but not exclusive, application in controlling robots in an educational environment, presenting information such as stories using the robot which performs a scene of operation comprising sound and action, where execution of the sound and action is executed in a coordinated format. The action of the robot may comprise robotic motion and an animation on a display of the robot which can be used to convey facial expression.

There have been numerous developments in the field of robotics, including robotics in an educational context. Several robots are commercially available which can execute motions coordinated with music. To name but a few, these models include QRIO, AIBO and Manoi. These robots offer to users pre-programmed systems which are programmed either by the manufacturer or the user. Options for customisation of the robot operation are somewhat limited. For instance, these systems typically require a lengthy and painstaking programming process if a user wishes to teach the robot to dance or to tell a story not pre-programmed into the robot. Typically, the user is required to use the respective manufacturer's software on a PC, recording the motion of the robot by stages, time the motion by synching it with the music the robot will be playing (typically stored on the robot), and then download just the action onto the robot.

Further, the amount of scenes of operation for these robots – i.e. sequence of movements – may be extremely limited.

The invention is defined in the independent claims. Some optional features of the invention are defined in the dependent claims.

Implementation of the techniques disclosed herein offer several significant advantages. For instance, creation of the new content file format for a file for downloading to the robot/robot controller, the content comprising both action and sound content components, provides the user with the opportunity to download new content for the robot, executable on the robot by the robot controller in a coordinated fashion. The robot controller, whether integral with, or separate from, the robot, decodes instructions within the content and controls the execution of the respective content components (action and sound) so that the sound and action are executed in a coordinated format.

Furthermore, new content for the robot may be downloaded at a user's convenience without the user having to go through the tedious program of training the robot to perform new content.

Provision of a new robot controller architecture also offers technical advantages. For instance, use of a distributed (or "matrix") processor is advantageous in that, because control of specific functions are grouped and processed by dedicated or semi-dedicated processors, these processors may be programmed to operate in parallel and, thus, lower-specification processors can be used, thereby requiring significantly lower power consumption when compared with industry standard processors in contemporary robotics having processing capabilities comparable to those of desktop computers.

One of the major concepts behind the techniques disclosed herein is that it may encourage end users, particularly children, to interact with a robot in an educative, fun and in a way which is appealing to the user. The techniques disclosed herein may allow for an integrated learning process for the child, bringing teaching content

through robotics. It may allow a child to learn about robotics, where the child can learn to sign and dance, mimicking the robot's operation. Further, the child can issue commands to the robot directly, with an immediate feedback seeing the robot operate.

Other benefits will become apparent from the following description of the invention, described by way of example only, and with reference to the accompanying drawings in which:

Figure 1 is a block schematic diagram illustrating a first control unit for controlling a scene of operation of a robot;

Figure 2 is a process flow diagram illustrating a process for controlling a scene of operation of a robot utilising the control unit of Figure 1;

Figure 3 is a block schematic diagram illustrating a content file for controlling a scene of operation of a robot;

Figure 4 is a process flow diagram illustrating a second processor for controlling a scene of operation of a robot utilising the control unit of Figure 1;

Figure 5 is a block schematic diagram illustrating operation of the control unit of Figure 1 to control of a scene of operation of a robot utilising the content file of Figure 3;

Figure 6 is a block schematic diagram illustrating a first apparatus for creating a content file for controlling a scene of operation of the robot;

Figure 7 is a block schematic diagram illustrating a second apparatus for creating a content file for controlling a scene of operation of a robot;

Figure 8 is a block schematic diagram illustrating a third control unit for controlling a scene of operation of a robot; and

Figure 9 is a block schematic diagram illustrating a fourth control unit for controlling a scene of operation of a robot.

Central to the techniques disclosed herein is the creation of a new file format, for a content file readable by a control unit or other apparatus for controlling a robot. A

content file in accordance with the new file format comprises content having commands for controlling a scene of operation of a robot. The content file has content components for both action and sound of the robot, where one of the content components provides a reference for execution of the other of the content components. The action content may be movement of the robot in some way, including a sequence of movements and/or an animation to be displayed on a display of the robot so as to convey, for example, a facial expression for the robot.

The overall logic implemented is fundamentally rather simple. Firstly, data/content (e.g. motion, sound and expression) is created. Secondly, this data/content is merged into a content file in accordance with the new file format, for downloading to a robot or robot control unit. Thirdly, commands in the data/content are executed/processed by the robot or robot control unit so that the action and sound elements of the scene of operation are executed in a coordinated manner.

One way to create the content includes the merging of individual content files for action and sound. For instance, if the action of the robot comprises movement of robot parts (such as arms or legs), then motion script for the robot servomotors is included in the action component of the content file. If the action of the robot comprises, say, display of an animation on a display of the robot, then expression script for facial features such as eyes is included in the action component of the content file. Of course, the content file may comprise of both motion and animation/expression content. Sound content may comprise of, say, voice (for storytelling, singing and the like) and/or music.

The content may be directed to a particular demographic of users. For instance, the content may be provided for the purpose of educating small children between the ages of 4 and 10 and/or other age groups.

Turning now to Figure 1, a first control unit for controlling a scene of operation of a robot is illustrated. Control unit 100 comprises a processor 102, a memory such as a RAM or an EEPROM device stores instructions executable by the processor for controlling execution of the scene of operation. Control unit 100 also comprises a storage 110 for storing content file 112 having sound component 114 and action component 116. In the example of Figure 1, storage 110 is provided as part of the control unit 100 but it will be appreciated that the storage 110 may be provided separately from control unit 100, for control unit 100 to communicate (i.e. read from and/or write to) storage 110. Storage 110 should be of sufficient size for allowing a user to storage therein multiple content files. Thus, storage 110 should offer 16GB or more of storage.

Control unit 100 may also comprise an I/O module 118 for receiving/downloading content file 112 and communicating with robot 122, which will be discussed shortly. Control unit 100 may also comprise a user interface 120 such as a touch display panel to allow a user to control the robot manually with commands only touch panel. This is particularly beneficial when control unit 100 is provided integral with robot 122. The user interface may also comprise of a device for entering commands into the control unit 100, such as a basic graphic user interface.

As noted, control unit 100 communicates with robot 122 over communications link 123. Robot 122 has arms 124, legs 126 and a head 128 having a display unit 130 for conveyance of facial expression and a speaker 132 for emitting sounds such as music or voice.

Thus, in essence, Figure 1 illustrates a control unit 100 for controlling a scene of operation of a robot 128. The scene of operation comprises a sound element, such as a poem or a song, and an action element, such as movement of a robot part 124, 126, 128 and/or display of an animation on display 130 of a facial expression of the robot. Control unit 100 reads commands for the scene of operation from content file

112 having a sound component 114 and an action component 116. Control unit 100 comprises processor 102 and a memory 104. Control unit 100 is configured, under control of processor 102, to execute instructions 106 stored in the memory 104 to read commands for the sound element of the scene of operation from the sound component 114 of content file 112 and to read commands for the action element of the scene of operation from the action component 116 of content file 112. Control unit 100 is also configured to control execution of a first one of the sound element and the action element and to control execution of a second one of the sound element and the action element with respect to execution of the first one thereof. That is, one of the sound and action elements of the scene of operation is used as a reference for the execution of the other of these. Thus, coordinated execution of the sound element and the action element of the scene of operation may be achieved by the control unit reading commands in the content file. The content file may be downloaded from, say, web servers either directly into storage 110 or via a user's computing device (not shown) for subsequent loading into storage 110. This provides a convenient and user-friendly system for executing customised content by the robot and allowing the user to obtain in a very convenient manner fresh content for the robot to execute.

It will be appreciated that the content file may be downloaded in its entirety into storage 110 prior to execution/reading by the control unit 100 or it may be streamed to control unit 100 for execution "on the fly" over a wireless link (not shown).

It will also be appreciated that the control unit 100 may be provided as an integral part of the robot 128, or separately as illustrated in Figure 1. If provided separately, control unit 100 may be fitted to the robot in a supplier's workshop or factory prior to shipping to a retail outlet or control unit 100 may be sold separately for retrofitting to an existing robot 128 by a user or qualified technician.

Content file 112 may have more components than sound component 114 and action component 116 as illustrated in Figure 1. A more detailed discussion of a content file is given with respect to Figure 3.

Turning to Figure 2, a first process for controlling a scene of operation of robot 128 by control unit 100 is illustrated. Figure 2 illustrates the principal steps only; a more detailed process is illustrated in Figure 4.

Process 200 of Figure 2 starts at 202. At step 204, control unit 100 reads commands for the sound element of the scene of operation from the sound component 114 of content file 112. At step 206, control unit 100 reads commands for the action element of the scene of operation from the action component 116 of content file 112. At step 208 (shown only as a single step in Figure 2 for the sake of convenience,) control unit 100 controls execution of the action element and the sound element, using one of these as a reference for the execution of the other. This is discussed in more detail with reference to Figures 4 and 5.

It will also be appreciated that it is not necessary for the steps as illustrated in Figure 2 to be repeated in the order illustrated.

As noted above, either the sound component/sound element or the action component/action element can be used as the reference. Use of the sound component as the reference may be particularly advantageous so that execution of the action content of the scene of operation is controlled with respect to the sound content. This is discussed in more detail with reference to Figures 3, 4 and 5. In the alternative, the action content may be used as a reference for controlling execution of the sound content/the sound element of the scene of operation. However, in doing so, it may be necessary to use synthesised sound as execution of the synthesised sound can be varied readily. When using sound as the reference, this provides a particularly advantageous solution as the sound may be played in a

continuous sequence and the sound content is readily usable -- as will be discussed in further detail below -- to provide the reference against which execution of the action element of the scene of operation is controlled.

Figure 3 illustrates a content file. Looking first at Figure 3a, content file 300 comprises principally one or more action components 304a, 304b and sound component 306. That is, a content file 300 for controlling a scene of operation of a robot comprises a sound component 306 comprising sound content for the scene of operation and an action component 304a, 304b comprising action content for the scene of operation. In the example of Figure 3, content file 300 comprises a header portion 302 and two action components 304a, 304b. In Figure 3a, action component 304a is a motion component comprising commands for controlling a sequence of movements of the robot and action component 304b comprises animation commands for controlling display of an animation on a display of the robot, such as display 130 of Figure 1. Such an animation may be used, for example, to convey a facial expression.

In this example, motion of robot 122 in the scene of operation is defined by how many frames the scene of operation is to be broken down into. The action component 304a may specify that the robot is to move from Point A to Point B, that the movement is to be completed within three seconds and that the movement is to be broken down into 30 frames, at 10 frames per second. Motion component 304a includes the motion commands for the scene which provides instructions to the servomotors of robot 122 to move. The scene information also includes information like:

- an identifier for each scene;
- duration of each scene;
- number of frames in the scene;
- an identifier for the servomotor of robot 122 which is to be moved for the scene;

- start position for the servomotor for the part 124, 126, 128 of robot 122 which is to be moved; and
- stop position for the servomotor for the moving part.

The scene define movements for robot 122, such as the movement of an arm 124 from one position to another, defining the start position and the end position, and the torque force the relevant servomotor on the robot is to apply, although it is possible to define a default torque value to ensure the robot never applies an excessive force in effecting a movement. A normal story -- i.e., as defined by a content file such as content file 300 -- can have as many as 700 to 900 scenes (or higher).

In creating the scene, the animator (the person performing the motion capture and creating the scene of operation) will first read through a story while listening to the voice over which forms the sound content of the content file. Then, the animator will formulate the types of body expression that best describe the content of each sentence. Once ready, the animator will capture each pose for the story, and into the relevant information -- positions, transition time, etc -- for the scene. The animator will also specify the length of time within which the scene is to be executed. This time duration is used by control unit 100 to divide the scene of operation into frames, which are sub-sections of the scene, thereby allowing the control unit to effect more precise control over execution of the scene of operation, and a means by which coordination of the execution of the action and sound components may be effected, as will be appreciated from the following discussion.

In this example, the animation content defines the commands for execution of an animation display on the display of the robot. The display may be a facial display and define, for example, facial expressions such as eye movements, and define the sequence of activation of pixels on a display such as display 130 on robot 122 for the

animation of the eye movements and the duration and sequence of pixel activation. So, the animation content may include information such as:

- a “start” code, to start the animation display
- The total number of expressions in the animation
- the first expression code (i.e. which pixels on the display are activated)
- the duration for the first expression
- and so on for the remainder of the animation sequence.

Over and above animations such as facial expressions, the animation content can also include commands for the display of displays such as indicating a low battery condition of the robot, a faulty servomotor and other suitable displays. The animation content might even include separate animation content file components to be executed as appropriate dependent upon the display, such as the facial expression, to be displayed on the display, such as happy, sad or neutral facial expressions each of which is defined in a separate file or file part.

The content file will comprise of data for:

- a “start” command;
- total number of scenes;
- total number of frames for the or each scene;
- total duration for the or each scene;
- start and stop positions for each scene; and so on.

In this example, the sound component 306 of content file 300 is a WAV file converted to OGG format. The WAV file can comprise of any recordable sound, music, voice, or any other noise but will typically comprise of voice (storytelling, poetry recital) or a mixture of voice and music (e.g. singing). Sound for the sound component 306 can be captured using PC sound recording software such as AudaCity. When the sound file uses voice content, the pitch of the captured voice

signal can be modified so that it sounds more child-like to appeal to child users. For example, in one implementation, it is preferred that the benchmark voice approximates to that of a five-year-old child. The sound file is then converted to OGG Vorbis format for further processing when merging with other media.

Sound component 306 of content file 300 is typically comprised of 512 byte blocks. Figure 3b illustrates a large-scale representation of the content file 300. In Figure 3b, action component 304a comprises commands defining a sequence of movements for a robot. Action component 304a is made up of action scenes, as defined above. Sound component 306 of content file 300 is comprised of a sound blocks 310 which, as noted above, are each 512 bytes in size. The significance of the action scenes and the sound blocks 310 will become apparent from the following discussion of Figures 4 and 5.

Figure 4 illustrates a second (more detailed) process for controlling a scene of operation of a robot utilising the control unit of Figure 1. As noted above, control unit 100 can use either an action component 304a, 304b of the content file 300 or a sound component 306 of the content file 300 (or the commands comprised within these components) as the reference to ensure coordinated execution of the two components, but the example of Figure 4 is given in the context of the sound component 306 of content file 300 being used as the reference. Process 400 starts at step 402. At step 404, control unit 100 obtains the size of sound component 306 of content file 300 and the duration of play, i.e., the length of time of play of the sound component. Control unit 100 can derive this information by scanning sound component 306 and reading the relevant information. In an alternative arrangement, this information may be embedded in header component 302 easily readable by control unit 100. At step 406, control unit 100 obtains or reads the number of sound blocks 310 of sound component 306. This can be done either from control unit 100 scanning sound component 306 to read the number of sound blocks 310 therein, or by a relatively simple arithmetic calculation in which the number of

sound blocks is calculated from a division of the total file size by the known block size which, in this example, is 512 bytes. In a further alternative arrangement, the number of sound blocks is written into the header component 302 for easy reading by control unit 100.

This principle applies equally for the other calculations performed by control unit 100 in the process of Figure 4. It may be possible to reduce the processing burden on control unit 100 by embedding the required information for calculations in the process of Figure 4 in the header 302 of content file 300. If the required information is embedded in the header 302, then control unit 100 is configured to derive information relating to the sound element and the action element from the header.

At step 408, the duration of play of a sound block is calculated by control unit 100 (or extracted from header 302 when embedded therein). If calculated by control unit 100, this is derived from the total duration of play of sound component 306 divided by the number of sound blocks 310 in sound component 306 as derived at step 406.

So, for example, if we assume the file size of the sound component 306 is 13 MB and the total duration of play is 9 minutes, then the total number of blocks equals 13 MB/512 bytes = 26,000 blocks (rounding up to the nearest thousand for the sake of simplicity). 9 minutes is equivalent to 540,000 ms so, the duration of play of a sound block is calculated from 540,000 ms/26,000 sound blocks = 20.7692 ms (approximately).

At step 412, control unit 100 obtains the duration of and number of frames in a scene of operation, "Scene 1", of the robot. In the present example, Scene 1 has a duration of 1000 ms and, say, 26 frames. The motion action -- e.g. the movement from Point A to Point B -- is sub-divided into 26 frames by control unit 100 for more precise control of the motion execution and also for the purposes of coordinating execution of the motion and execution of the sound component. The scene may be

divided into frames of equal duration; for example, if 26 frames are to be executed in a 1000 ms scene, then the scene is divided into 26 frames of 38.46 ms duration. Alternatively, the scene of operation may be divided into frames of unequal length.

So, as Scene 1 must complete within 1000 ms, the number of sound blocks for a 1000 ms scene corresponds with a total of 1000 ms/20.7692 ms = 48 sound blocks 310 (approximately). This is calculated at step 414. Control unit 100 may allow for a "tolerance" frame of one or two so that, if the action completes one or two frames late, then that is acceptable and not considered to be loss of co-ordinated execution.

At step 416, a pulse sequence is derived, discussed in more detail with reference to Figure 5 which shows sound component 306 and action component 304a of content file 300 separated. As noted above, sound component 306 is comprised of sound blocks 310 and action component 304a is comprised of action frames 308 defining a sequence of movements for the robot, calculated as described above. The numbers of sound blocks and action frames have been simplified for ease of understanding when compared with the numbers of sound blocks and action frames discussed above with reference to Figure 4. In the simplified example of Figure 5, a scene of a particular duration comprises a total of 20 frames of movement 308 and 40 sound blocks 310. Control unit 100 derives the sequence of pulses from a number of sound blocks (i.e. 40) and a number of action frames (i.e. 20) for the scene of operation. Thus, if the scene of operation requires coordinated execution of a total of 20 frames of movement 308 and 40 sound blocks 310 within a specified time frame, then each action frame must execute within a total of two sound blocks. That is, we have two sound blocks 310 for one frame of action 308. Thus, control unit 100 controls execution of the scene of operation using a sequence of pulses related to the sound blocks 310, with the sound content being played continuously, thereby serving as a reference for the action content. This can be generalised in that if we have Y sound blocks in a scene of operation and we have X action frames in the scene of

operation, then a formula governing the time of execution of a particular frame is as follows:

[Xth * (Y/X)] – 1

where:

Xth is the frame number currently being executed

Y is the total number of sound block in this scene

X is the total number of frames in this scene.

Thus, in this example, each pulse is derived for the start of each second sound block.

When the number of sound blocks in a scene of operation is not an integer multiple of the number of action frames in the scene of operation, an approximation is made. For instance, in another example where there are 20 frames and 50 sound blocks in a scene of operation, a pulse is initiated after two sound blocks, but the next pulse is initiated after a further three sound blocks and the process is repeated over the duration of the scene of operation with a period of a pulse for two sound blocks followed by a pulse for three sound blocks. Control unit 100 is arranged to detect the onset 502 of each second sound block 310 and to generate a control pulse 504 therefor to initiate execution of each frame of action. That is, control unit 100 is configured for a pulse 504 of the sequence of pulses to correspond to a start of play of a sound block 310, the pulse 504 defining a reference for initiation of an action frame 308. When a pulse 504 is generated by control unit 100, a frame is then executed. The same or a similar technique can be implemented for an animation display where, when the display is required to change, say, the robot's facial expression, this can be synchronised with the sound blocks in the same or a similar way as the action frames are. For instance, if Display 1 has a duration of 400 msecs, then display 1 must execute in 400/20.7692 ms = 19 sound blocks.

Thus, at step 418, control unit 100 executes the scene of operation using the pulse sequence as a control reference.

In addition, control unit 100 monitors execution of the action element relative to execution of the sound element. Dependent upon detection that execution of the action element and execution of the sound element is in an unsynchronised state, control unit 100 modifies one of the action element and the sound element. As mentioned above, control unit 100 operates to control execution of the action element and the sound element of the scene of operation of the robot in a coordinated -- or synchronised -- manner. However, it is possible that synchronisation may be lost. If synchronisation is lost, control unit 100 operates to try to recover the situation.

If or when execution of one of the action element and the sound element lags behind the other, control unit 100 operates to adapt one of the sound element and the action element to recover the synchronised state of execution. For the purposes of the present discussion, and as noted above, in this example control unit 100 controls execution of the scene of operation of the robot using sound content 306 as the reference for the coordinated execution. Thus, if or when execution of the action element lags behind execution of the sound element, control unit 100 operates to adapt the action element. This is most likely necessary when the action element is the movement/motion of a robot part 124, 126, 128 in a scene of operation. Lagging of the motion component may occur if, say, a user holds onto a robot part, thereby inhibiting its motion. It can also occur in instances where the robot has a faulty servomotor or if there is a significant drop in robot battery charge.

Therefore, at step 420 control unit 100 monitors execution of the action element and the sound element to ensure that the coordinated execution is as planned. If execution is as planned -- that is, execution of the action element and the sound element is in a synchronised state -- then control unit 100 determines at step 421

whether the scene has been completed. If the scene has not been completed, process 400 loops back to step 420 for continued monitoring of the synchronised execution of the scene of operation. If at step 421 the end of the scene is detected by control unit 100, control unit 100 then checks at step 422 to determine whether there are further scenes of operation for execution. If more scenes are to be executed, process 400 loops back to step 410 to retrieve the next scene for execution. If no further scenes are to be executed, the process ends at step 430.

Returning to step 420, if an unsynchronised state is detected, control unit 100 tries to recover the situation. One way of doing this is for control unit 100 to skip one or more frames of action 308 as illustrated at step 424. Thus, if execution of the action element of the scene of operation lags behind the sound element by, say, two or three sound blocks, then control unit 100 skips two or three action frames (depending on the number of action frames the action element lags by), in an attempt to recover the synchronised state. Typically the control unit 100 will control execution of the scene of operation at a rate of around 30 frames per second. A human eye is capable of distinguishing about 15 frames per second. Thus, operation at a rate of around 30 frames per second appears to be very smooth operation to a human eye and the skipping of the few frames is likely not to be noticeable to a human user observing the robot 128 perform the scene of operation.

Of course, if the action element of the scene of operation is being used by control unit 100 as the reference for execution of the sound element, it would also be possible to skip the playing of a selected number sound blocks in order to recover synchronised operation. However, such a scenario is less likely and, in any event, skipping of sound blocks may be more noticeable to a human user.

The determination step at 420 may operate according to a user-defined threshold for the number of action frames which are lagging behind playing of the sound blocks. For instance, the animator of the scene of operation may determine that a

lag of two frames of action behind the execution of the sound component is unacceptable and recovery is necessary, or, alternatively, a lag of three frames of action behind the sound component is deemed to be unacceptable.

As an alternative to skipping frames of action from the scene of operation, control unit 100 may operate to speed up the execution of the action frames. However, this may be less desirable given that it would require an increase in torque force applied by the servomotor.

After skipping frames at step 424 (or otherwise adapting the action element or the sound element) at step 426, control unit 100 determines whether coordinated/synchronised execution of the scene of operation has been recovered. If a synchronised state has been recovered, process 400 loops back to step 420 for continued monitoring of synchronised execution. If however a synchronised state of execution has not been recovered at step 424 -- for instance, the user may have retained hold of the part 124, 126, 128 of robot 122 thereby inhibiting its further motion or if the battery of the robot is in a significantly discharged state -- control unit 100 skips to the end of the scene at step 428.

Process 400 continues to step 422 where, as mentioned above, control unit 100 checks to determine whether further scenes of operation are to be executed.

When one scene of operation has been completed and control unit 100 is retrieving the next scene of operation for execution, the necessary calculations for the next scene of operation are made to synchronise the number of frames in the scene of operation with the number of sound blocks in the scene. Control unit 100 recalculates the total number of blocks of time per frame as discussed above. This is necessary because transition from one scene of operation may require a change from, say, 10 frames of action in a duration of 30 seconds to 20 frames of action in a duration of 20 seconds.

It will be appreciated that the example given above has been discussed in the context of the action element of the scene of operation being a movement or motion of the robot. However the techniques disclosed are equally applicable for other types of action.

Returning to Figure 3, content file 300, as noted, comprises of header 302, action components 304a, 304b and sound component 306. The overall size of content file 300 may be of the order of 14 MB or 15 MB. Typically, motion component 304a will be of the order of 50 kB or so. Animation/expression component 304b will be of the order of around 4 kB. The remainder of content file 300 is, in this example, taken up by sound component 306 which may be of the order of 13 MB or so.

Header component 302 comprises information relating to the respective sizes of the components of content file 300. Additionally, header component 302 may comprise information relating to the number of action scenes or action frames 308, and the start and end positions thereof, in motion component 304a, the number of sound blocks 310 and the start and end positions thereof in sound component 306 and, if applicable, the number of animation frames in animation/expression component 306b. Header component 302 may also comprise pointers/flags pointing to locations in the respective components 302, 304a, 304b, 306 of content file 300 relating to individual action frames, sound blocks etc., for the start of the frames and the sound blocks and so on.

Figure 6 illustrates an apparatus for creating a content file for controlling a scene of operation of a robot, such as the content file of Figure 3. Apparatus 600 comprises microprocessor 602, a memory -- such as a RAM or an EEPROM -- 604, for storing instructions executable by processor 602. Apparatus 600 comprises a storage 610 for storing a content file 612 comprising at least an action component 614 and a sound component 616. Apparatus 600 comprises write module 608 for writing the

respective components to content file 612. Peripheral devices in apparatus 600 may include an I/O module 618 for interfacing with external devices and/or a user interface which may comprise of a graphical user interface.

In essence, Figure 6 illustrates an apparatus 600 for creating a content file 612 for controlling a scene of operation of a robot. Apparatus 600 comprises a processor 602 and a memory 604. Apparatus 600 is configured, under control of processor 602 to execute instructions 606 stored in the memory 604 to write sound content for the scene of operation to a sound component 616 of the content file 612 and to write content for the scene of operation to an action component 614 of the content file.

Apparatus 600 may create content file 612 where the action content comprises movement commands written in to the action component 614 for controlling a sequence of movements for the robot. The action content may also comprise animation commands for controlling display of an animation on a display of the robot such as to convey a facial expression for the robot.

Apparatus 600 is configured to derive action header information from the action content 614 and to write the action header information to a header component (not illustrated in Figure 6) of the content file 612. The header component may comprise of the information for motion and/or animation as discussed above. Additionally or alternatively, apparatus 600 derives sound header information from the sound content and to write the sound header information to the header component of the content file.

Turning now to Figure 7, a second apparatus for creating a content file for controlling a scene of operation of a robot is illustrated. Apparatus 700 comprises a first processor 702 having an associated EEPROM or other memory 704. EEPROM 704 is of course a non-volatile memory storing instruction code for the operation of first processor 702. Clock 706 is provided for time-stamping the created file or

components thereof. SDRAM or other volatile memory 708 is provided for temporary loading of instruction code for execution by first processor 702. Encryption module 710 is provided for encrypting a content file, or a component thereof, created by apparatus 700. uSD (Micro-SD) socket 711 allows a uSD memory card to be inserted therein. The uSD memory card can be provided for loading of the content file created by apparatus 700 thereon for reading of action/sound/animation content into apparatus 700 for creation of the content file.

Apparatus 700 also comprises a second processor 712 having an associated EEPROM or other non-volatile memory 714 storing instruction code for operation of second processor 712. Ethernet module 716 and Ethernet UTP connector 718 allows apparatus 700 to communicate over a computer network. VGA connector 720 is provided for connection of a display and connector 722 is provided for connection of a mouse or other human interface device to facilitate user interface with apparatus 700. Apparatus 700 is powered by a power circuit 724.

In operation, individual media files -- for the action (whether motion or animation/expression), and sound -- are saved to a uSD card inserted in uSD socket 711. To generate a content file using apparatus 700, the motion content is read and the relevant information pertaining to the motion content is formulated to an array for writing to the header component. The sound file, which may be in OGG format, is read and the relevant information pertaining to the sound content is formulated to the array for writing to the header component. Further, apparatus 700 reads the display file (for the display of the animation on a display of the robot) and the relevant information relating to the animation content is formulated to an array for writing to the header component.

The content being written to the content file -- whether sound or action content, or the header component -- is written using a linear writing process where one block is written after another. The array list is formatted into 512 byte blocks and then

stored to SDRAM 708. Media content is then written to 512 byte blocks and stored in SDRAM 708. Encryption is applied to the files stored in SDRAM 708 using encryption module 710 and the content file, thus encrypted, is stored in the uSD memory card inserted in uSD socket 711 after which it may be uploaded to a web server for a user to download

The file format definition of the merged file may be as illustrated in Table 1.

Category	Byte Location	Data Store (Hex)	Item Name	Size (Byte)	Description
	0000		Identifier	4	"MTYF"
	0004		Media Types	1	Total number of media types in file, max 255
	0005		Year created	1	256 years from year 2000
	0006		Time created	4	In seconds within a year
Header	0010		Year modified	1	256 years from year 2000
Total Size is	0011		Time modified	4	In seconds within a year
256 Byte	0015		Creator ID	1	ID code of file creator
	0016		Creator Loc	1	Location of file creator
	0017		File size	4	Actual size of physical file, not size on disk
	0021		Checksum	4	Checksum value AFTER encryption
	0025		Format Version	3	1st byte => Major, 2nd & 3rd Byte => Minor in word-size
	0028		Unused	1	Unknown
	0029		Unused	1	Unknown
	0025		Unused	231	{JUMP}
	0256		1st Media Type	1	Type codes from 0x00 ~ 0xFF
Type Code	0257		2nd Media Type	1	:
of Media	0258		3rd Media Type	1	:
Total Size is	0259		Unused	1	:
256 Byte	0260		Unused	1	:
	0261		Unused	251	{JUMP}
	0512	00 00 04 00	Pointer to 1st	4	Start Address of 1st Media Type
	0516		Pointer to 2nd	4	Start Address of 2nd Media Type
Pointer to	0520		Pointer to 3rd	4	Start Address of 3rd Media Type
Media	0524		Unused	4	:
Block	0528		Unused	4	:
Total Size is	0532		Unused	4	:
512 Byte	0536		Unused	4	:
	0540		Unused	4	:
	0544		Unused	480	{JUMP}
	1024		:		
	:		:		
Media	:		:		
Blocks	:		:		
	:		:		

Table 1

Once the content has been merged into the content file, such as the content file of Figure 3, and encrypted it is transferred to web servers for users to download therefrom. Once a user makes a selection of the media content to download from the web server, an application synchronises the user's account on the server with information stored at the user's computing device by the application.

The apparatus 700 also performs encryption of the content file using, say, SHA-256 encryption. Apparatus 700 may also be arranged to generate a random challenge key for use by the authentication chip in a control unit such as that of Figure 9, discussed below, As such, a content file may also have a text file either associated with it, or have a text portion within it for use as the challenge key. The file (or files) can be downloaded and saved to storage medium for the robot. When the user selects to play the content, the random challenge text content file are sent to the authentication chip decryption and if decryption is successful and the file validated, the content will then be sent to the necessary engines for execution.

Figure 8 illustrates a third control unit for controlling a scene of operation of a robot 810. Figure 8 illustrates a new distributed -- or “matrix” -- architecture for a robot controller. Control unit 800 comprises, principally, a matrix controller for controlling sound and action of the robot and comprising a sound controller 802 for controlling sound of the robot and an action controller 804 for controlling action of the robot. In the example of Figure 8, the matrix controller further comprises a root controller 806 for providing a control function for the sound controller 802 and/or the action controller 804. The control function will be described in greater detail below with respect to Figure 9 but, in summary, one type of control function which of its controller 806 can provide for either or both of sound and action controllers 802, 804 is to check for, and to load, any firmware updates for these controllers.

Control unit 800 may also comprise an I/O module 808 for interfacing with external devices, such as robot 810.

Turning now to Figure 9, a more detailed view of a control unit having a matrix controller for controlling a scene of operation of a robot is illustrated. Control unit 900 comprises principally a sound controller 904 for controlling sound of the robot and an action controller 906 for controlling action of the robot. Action controller 906 may be arranged to control movement of the robot. Control unit 900 manages this,

under control of action controller 906, to extract data relating to motion content in a content file, the motion content comprising motion commands for a scene of operation of the robot and to communicate the data relating to motion content to the action controller 904. Thus, motion commands (or pointers to the addresses of the commands in the action component of the content file) are extracted from the downloaded content file and sent to action controller 904 for execution, optionally via root controller 902, discussed below.

Additionally or alternatively, the action controller 906 may be configured to control display of an animation on a display of the robot. So, control unit 900 extracts, under control of the sound controller 904, data relating to animation content from the content file, the animation content comprising animation commands for a scene of operation of the robot, and communicates the data relating to animation content to the action controller 904. Thus, the animation display commands such as commands for animation of a facial expression for the robot (or pointers to the addresses of the commands in the action component of the content file) are extracted from the downloaded content file and sent to action controller 904 for execution again, optionally, via root controller 902.

In the example of Figure 9, control unit 900 also comprises root controller 902 which, amongst other things, provides a control function – such as the checking for, and loading of, firmware updates – for the sound controller 904 and/or the action controller 906.

Yet further, in some implementations, control unit 900 is provided with a fourth controller 908 for a remote control function of the robot. In this example, control unit 900 is arranged to be controlled remotely via Bluetooth connector 926 for interfacing with a remote control unit (not shown) communicating over a Bluetooth link (also not shown).

Any of the controllers 902, 904, 906, 908 are in this arrangement microcontroller class, such as the Parallax Propeller (P8X32A-Q44) processor. Each processor 902, 904, 906, 908 comprises of 8 individual cores. Therefore, this matrix MCU configuration has a total of 32 cores.

Root controller 902 is the main CPU that boots with a boot loader firmware installed from non-volatile memory 910 which, in the example of Figure 9, comprises an EEPROM. The boot loader firmware will be discussed in more detail below. Control unit 900 has a root non-volatile memory 912 associated with the root controller 902 for storing predefined recovery codes for sound and action for operation of the robot in a recovery mode. Thus, if the robot is unable to function fully -- say if it is unable to perform a scene of operation in accordance with the techniques discussed above -- then it may still be able to operate in a recovery mode to inform a user of its limited functionality by, for example, emitting a sound message such as "low battery", “faulty servomotor” or the like.

As mentioned above, non-volatile memory 910 is loaded with a boot loader firmware for controlling root controller 902 to perform the following, normally at bootup of the control unit/robot:

- the root controller 902 checks its associated storage in either a uSD inserted in uSD socket 922 or NAND flash memory 920 to determine if there are any firmware upgrades for any new firmware to upgrade root non-volatile memory 910. The boot loader firmware checks for which of these storage devices is in operation. If root controller 902 detects any new firmware updates, controller 902 will flash (read new firmware, erase old firmware in root non-volatile memory 910 and write the new firmware into root non-volatile memory 910). After that, root controller 902 erases the new firmware file in its storage.
- Root controller 902 then checks to determine whether a firmware upgrade is available for sound controller 904 and associated non-volatile memory 914,

which stores predefined sound codes for the robot. If the root controller 902 detects a firmware upgrade for sound controller 904, root controller flashes (reads the new firmware, erases the old firmware in sound controller 904 and writes the new firmware into action controller 904). After that, root controller 902 erases the new firmware file in its storage.

- Root controller 902 continues to do this for action controller 906 and associated non-volatile memory 916 which stores predefined action codes (such as, say, the wave of an arm) for the robot. If appropriate, root controller continues with this process for fourth controller 908 and its associated non-volatile memory 918.
- Once all is done, root controller 902 will run its firmware (current or newly flashed) from non-volatile memory 910.

As noted above, provision of this new robot controller architecture is advantageous in that, because control of specific functions are grouped and processed by dedicated or semi-dedicated processors. Thus, for instance, action controller 906 controls all action – e.g. motion, touchpad control – functionality. Action controller 906 sends commands to all the robot servomotors for robot movements and display commands for, say, animation of facial expressions on the robot display.

These distributed processors may be programmed to operate in parallel and, thus, lower-specification processors can be used, thereby requiring significantly lower power consumption when compared with industry standard processors in contemporary robotics having processing capabilities comparable to those of desktop computers. For instance, any or all of processors 902, 904, 906, 908 may have a processing speed of, say, 80 MHz. However, processor 908 may operate at an increased clock speed if necessary for performing the desired remote control function over Bluetooth or, say, USB at 96Mhz or thereabouts.

Non-volatile memory 914 is split into two tiers. The lower tier comprises the main application under the upper tier comprises a lookup table. This non-volatile memory does not require a loader. If any flashing of its EEPROM is required, it is the lower tier which is flashed. Sound controller 904 manages the SDRAM and OGG decoder 924. That is, control unit 900 extracts, under control of sound controller, sound content from the content file which comprises sound for a scene of operation for the robot. This look up table has pre-programmed sounds such as "hello", for the robot to emit.

Non-volatile memory 918 is split into two tiers. The upper tier includes a look up table for communicating between the robot and the (say, Bluetooth) remote controller and includes application commands such as : play media; switch to remote control, etc.

In the current example, the non-volatile memories 910, 912, 914, 916, 918 are EEPROMs of the type provided by Microchip, 24LC512 serial EEPROM which is a 512k-bit size chip. The lower 32k-byte partition of non-volatile memory 910 stores the boot loader firmware. The upper 32k-byte partition will store the main firmware code. Non-volatile memory 912 is mainly used for memory storage for frequently used symbols & data and for recovery codes as discussed above. For each of non-volatile memories 914, 916, 918, the lower 32k-byte partition stores the active firmware for their respective controllers 904, 906, 908. The upper 32k-byte partitions will be used as memory storage by its respective IC.

Clock 928 is, in this example, a Maxim DS1307+ Real-Time Clock. Clock 928 is tied to root controller 902 and is mainly used by the SD module for root controller 902. Data for clock 928 is pulsed to the other controllers whenever necessary via a InterMPComm module. The InterMPComm module is provided in all of the controllers and is used for inter-controller communication.

Micro-SD (uSD) socket 922 houses a uSD card which conforms to SD industry standard. The socket is, in the current example, the Hirose model DM3BT-DSF-PEJS which is a push-push for better hold on uSD card.

NAND Flash module 920 is a high-speed NAND Flash model that conforms to, for example, either Serial Quad I/O or ONFi standard.

For power circuit 930 in the current example, the input AC voltage is 12v, 5A. It is regulated to 10v for use by robot servo motors and then regulated to 5v & 3.3v by two voltage regulators (not shown, but of the types LM2940 & LM2937). Power circuit 930 may use a MOSFET (ZXMN2B03E6) and provides a signal to action controller 906 for charging NiMH (Nickel-Metal Hydride) batteries by measuring voltage decay duration.

Accelerometer module 932 is in this example an Hitachi H48C, 3-Axis accelerometer module, that senses gravitational (g) force of +-3g on three axes (X, Y & Z). Action controller 906 can be configured to poll constantly and post to the other controllers via InterMPComm.

Control unit 900 also comprises a display connector 934 and is, in this example, from 4D uOLED 96-G1-SGC. The display unit is mounted on the head unit (not shown) of the robot (also not shown). The display unit is a 96x64 pixel resolution with 65k true to life colours. Communication with action controller 906 is via UART.

Controller 900 also comprises four servo connectors 936a, 936b, 936c, 936d which are mounted to service 16 servo motors of the robot. The robot servomotors are wCK 1108k & 1111k models in this example, which are intelligent module with a built-in controller from Atmel. All motion commands are issued directly from action controller 906.

Controller 900 may also comprise a 32MB SDRAM chip, such as the MT48LC32M8A2 model, from Micron. This is mainly used as buffer storage for sound decoding and decrypting files.

OGG Decoder 924 decodes the sound component within the encrypted content file downloaded to control unit 900 which is encoded using OGG Vorbis. Therefore the OGG decoder IC, VS1000d, from VLSI.fi in this example is used. This is a system-on-a-chip OGG Vorbis player which can be function as a standalone IC.

Op-Amp 938 is, in the example of Figure 9, a TS4998 differential input stereo audio amplifier IC to amplify the audio sounds (voice, music etc) to two speakers (0.5w, 8ohm) that will be connected to the speaker connectors 940a, 940b.

Exemplary Crypto-Chip 942/Crypto-Memory 944 is used to encode the digital media content files using SHA-256 enabled by this AT88SC series crypto-chip from Atmel. All content files will be decrypted by blocks of 512-bytes and stored inside SDRAM before being dispatched out to other modules for processing. AT88SC chip will store the secret key that is placed during production.

Control unit 900 may implement a software version of USB 1.1 host controller (12Mb/s) on fourth controller 908 for the Bluetooth unit. Therefore, the Bluetooth controller can be of relatively low specification for connection to the control unit 900 enabling it to act as a Bluetooth host controller serving out SPP (Serial Port Profile) service. This allows communication with the robot command and control application running on the user's computer device (not shown). This application is used to synchronise all purchases made on the web server site, configurations and robot selections (play media, switch to autonomous or remote controlling the robot).

Touch connector 946 allows connection of a touch connector such as Freescale's MPR121 IC which is a proximity capacitive touch sensor controller. Touch connector

946 is connected to action controller 906 via an I2C bus. It acts as motion sensor for the robot and provides simple hand gesture motion on the robot's chest (when the controller is installed integral with the robot) as controls for volume and other emotional expressions to the robot by the end-user.

It will also be appreciated that other content – ie content not in the format of the content files as disclosed herein – such as MP3 files can be downloaded and played on the robot under control of the control units.

It will be appreciated that the invention has been described by way of example only. Various modifications may be made to the techniques described herein without departing from the spirit and scope of the appended claims. The disclosed techniques comprise techniques which may be provided in a stand-alone manner, or in combination with one another. Therefore, features described with respect to one technique may also be presented in combination with another technique.

CLAIMS

1. A control unit for controlling a scene of operation of a robot, the scene of operation comprising a sound element and an action element, the control unit being configured to read commands for the scene of operation from a content file having a sound component and an action component, the control unit comprising:

a processor; and

a memory; wherein

the control unit is configured, under control of the processor, to execute instructions stored in the memory:

to read commands for the sound element of the scene of operation from the sound component of the content file;

to read commands for the action element of the scene of operation from the action component of the content file;

to control execution of a first one of the sound element and the action element; and

to control execution of a second one of the sound element and the action element with respect to execution of the first one of the sound element and the action element.

2. The control unit of claim 1 configured to monitor execution of the action element relative to execution of the sound element, and dependent upon detection that execution of the action element and execution of the sound element is in an unsynchronised state, to modify one of the action element and the sound element.

3. The control unit of claim 1 or claim 2, wherein the sound component of the content file comprises of sound blocks, and the control unit is configured to control execution of the scene of operation using a sequence of pulses related to the sound blocks.

4. The control unit of claim 3 configured for a pulse of the sequence of pulses to correspond to a start of play of a sound block, the pulse defining a reference for initiation of an action frame.

5. The control unit of claim 4 configured to derive the sequence of pulses from a number of sound blocks for the scene of operation and a number of action frames for the scene of operation.

6. The control unit of any preceding claim configured to process the content file where the action element of the scene of operation comprises of a sequence of movements for the robot and the action component of the content file comprises commands for controlling the sequence of movements.

7. The control unit of any preceding claim configured to process the content file where the action element of the scene of operation comprises an animation for display on a display of the robot and the action component of the content file comprises commands for controlling display of the animation.

8. The control unit of any preceding claim configured to derive information relating to the sound element and the action element from a header of the content file.

9. Apparatus for creating a content file for controlling a scene of operation of a robot, the apparatus comprising:

a processor; and

a memory; wherein

the apparatus is configured, under control of the processor, to execute instructions stored in the memory:

to write sound content for the scene of operation to a sound component of the content file; and

to write action content for the scene of operation to an action component of the content file.

10. The apparatus of claim 9 configured to create the content file where the action content comprises movement commands for controlling a sequence of movements for the robot.

11. The apparatus of claim 9 or claim 10 configured to create the content file where the action content comprises animation commands for controlling display of an animation on a display of the robot.

12. The apparatus of any of claims 9 to 11 configured to derive action header information from the action content, and to write the action header information to a header component of the content file.

13. The apparatus of any of claims 9 to 12 configured to derive sound header information from the sound content and to write the sound header information to a header component of the content file.

14. A content file for controlling a scene of operation of a robot, the content file comprising:

a sound component comprising sound content for the scene of operation; and

an action component comprising action content for the scene of operation.

15. The content file of claim 14, wherein the action content comprises movement commands for controlling a sequence of movements for the robot.

16. The content file of claim 14 or claim 15 wherein the action content comprises animation commands for controlling display of an animation on a display of the robot.

17. The content file of any of claims 14 to 16, wherein the content file comprises a header component having action header information derived from the action content.

18. The content file of any of claims 14 to 17, wherein the content file comprises a header component having sound header information derived from the sound content.

19. A control unit for controlling operation of a robot, the control unit having a matrix controller configured to control sound and action of the robot, the matrix controller comprising:

a sound controller for controlling sound of the robot; and

an action controller for controlling action of the robot;

20. The control unit of claim 19, wherein the matrix controller comprises a root controller configured to provide a control function for the sound controller or the action controller.

21. The control unit of claim 20 comprising a root non-volatile memory associated with the root controller, for storing predefined recovery codes for sound and action for operation of the robot in a recovery mode.

22. The control unit of any of claims 19 to 21, wherein the action controller is arranged to control movement of the robot.

23. The control unit of any of claims 19 to 22, wherein the action controller is configured to control display of an animation on a display of the robot.

24. The control unit of any of claims 19 to 23, wherein the control unit is configured to extract, under control of the sound controller, sound content from a content file, the sound content comprising sound for a scene of operation for the robot.

25. The control unit of any of claims 19 to 24, wherein the control unit is configured to extract, under control of the sound controller, data relating to motion content in a content file, the motion content comprising motion commands for a scene of operation of the robot, and to communicate the data relating to motion content to the action controller.

26. The control unit of any of claims 19 to 25, wherein the control unit is configured to extract, under control of the sound controller, data relating to animation content from the content file, the animation content comprising animation commands for a scene of operation of the robot, and to communicate the data relating to animation content to the action controller.

27. The control unit of any of claims 19 to 26 comprising a sound non-volatile memory associated with the sound controller for storing predefined sound codes for the robot.

28. The control unit of any of claims 19 to 27 comprising an action non-volatile memory associated with the action controller for storing predefined action codes for the robot.

29. A robot having a control unit according to any one of claims 1 to 8 or claims 19 to 28.

30. A method, implemented in a control unit, for controlling a scene of operation of a robot, the scene of operation comprising a sound element and an action element, the control unit being configured to read commands for the scene of operation from a content file having a sound component and an action component, the method comprising, under control of a processor of the control unit:

reading commands for the sound element of the scene of operation from the sound component of the content file;

reading commands for the action element of the scene of operation from the action component of the content file;

controlling execution of a first one of the sound element and the action element; and

controlling execution of a second one of the sound element and the action element with respect to execution of the first one of the sound element and the action element.

31. A method, implemented in an apparatus, for creating a content file for controlling a scene of operation of a robot, the method comprising, under control of a processor of the apparatus:

writing sound content for the scene of operation to a sound component of the content file; and

writing action content for the scene of operation to an action component of the content file.

32. A method for controlling operation of a robot, implemented in a control unit having a matrix controller, the method comprising:

controlling sound of the robot using a sound controller of the matrix controller; and

controlling action of the robot using an action controller of the matrix controller.

33. A computer readable medium having stored thereon the content file for controlling a scene of operation of a robot of any of claims 14 to 18.

34. A computer readable medium having stored thereon computer-readable instructions for executing, under control of a processing device, the method of any of claims 30 to 32.