https://doi.org/10.5392/IJoC.2019.15.4.016

An Effective Visualization of Intricate Multi-Event Situations by Reusing Primitive Motions and Actions

Jong Hee Park
School of Electronics Engineering
Kyungpook National University, Daegu, ASI|KR|KS002|TAEGU, Korea

Jun Seong Choi
School of Electronics Engineering
Kyungpook National University, Daegu, ASI|KR|KS002|TAEGU, Korea

ABSTRACT

The efficient implementation of various physical actions of agents to respond to dynamically changing situations is essential for the simulation of realistic agents and activities in a cyber world. To achieve a maximum diversity of actions and immediate responsiveness to abrupt changes in situations, we have developed an animation technique in which complex actions are recursively constructed by reusing a set of primitive motions, and agents are designed to react in real-time to abrupt ambient changes by computationally satisfying kinematic constraints on body parts with respect to their goals. Our reusing scheme is extended to visualize the procedure of realistic intricate situations involving many concurring events. Our approach based on motion reuse and recursive assembly has clear advantages in motion variability and action diversity with respect to authoring scalability and motion responsiveness compared to conventional monolithic (static) animation techniques. This diversity also serves to accommodate the characteristic unpredictability of events concurring in a situation due to inherent non-determinism of associated conditions. To demonstrate the viability of our approach, we implement several composite and parallel actions in a dynamically changing example situation involving events that were originally independent until coincidentally inter-coupled therein.

***Key words**: Reusing Motions and Actions, Recursive Motion Construction, Assembly-based Animation, Action Parallelism, Real-time Reaction, Kinematics, Virtual World, Action Animation.*

1. INTRODUCTION

The implementation of diverse physical actions of an agent in response to ever-changing situations is essential for simulation of realistic agents and situations in a cyber-world [1]. However, human agents in a realistic cyber-world reveal such enormous motion variability and action diversity that it is prohibitively time consuming to pre-author all their motions and actions. Further, abrupt changes in dynamically changing situations and coincidental interaction between autonomous agents require real-time reaction in the execution time. Among the motion animation techniques, key frame technique and motion capture technique can be compared with respect to visual realism, motion variability, authoring speed & cost, recording cost, data reduction, etc. [2]-[7]. As those existing research efforts focus on visual realism, they are limited in creating diverse actions or reacting to unexpected situations [8]. Motion capture techniques can achieve visual realism but their application tends to be limited to a small set of motion types due to high recording cost. Key frame techniques or interpolation-based techniques for improving data reusability are confined to reusing already captured image patterns in their entireties, but not flexible enough to synthesize new motions from those image data [2], [3]. Our approach to implementation of agent's motions and actions for a cyber-world is aimed to maximize the variability of motions and the diversity of actions at an expense of suboptimal visual realism. It is required to handle not just the diversity of intra-agent actions, but also countless interactive patterns of each inter-object action along the continuum of contact timings between agents or objects. These action diversity and motion variability requirements are further compounded by unforeseen situational changes in their execution and coincidental inter-agent interactions common in our target cyber world. These requirements are to be addressed with respect both to authoring scalability and real-time execution.

We first model the agent's body as a kinematic hierarchy with its body parts functionally interconnected via joints. Its movement is modeled by the kinematic equations of the

* *Corresponding author, Email: daegulink@naver.com*

Manuscript received Jun. 5, 2019; revised Nov. 21, 2019; accepted Nov. 25, 2019

hierarchy. These equations define the configuration of the hierarchy in terms of its joint parameters. Based on this kinematic model, we define a set of primitive motion in terms of parameterized motion functions instead of captured images [9], [10]. These functions are sequenced successively into more complex motion functions, and eventually into diverse action functions. This successive assembling method with reusable (schematic) motion functions drastically curtails the authoring cost compared to those conventional 'monolithic' authoring. Specifically, motions can be computationally generated under their associated kinematic constraints among body parts based on functionally interconnected joints (corresponding to control points in key frame) instead of specifying Knots on every real image newly captured. This computational generation in our technique allows key frames to be reflected on the motion function during the animation process for a body part, which is an essential real-time animation scheme for simulating an emergent situation involving coincidental inter-agent interaction.

An action in our model is either of an intra-object type or inter-object type, depending on whether its goal is motion itself or change in situation. An intra-object action, e.g., walk(), is animated by sequencing body parts' motions according to its predesigned procedure. Its changeable aspects are confined to its overall speed, occurrence timing, or other minor variations within the domain of each parameter variable. An intra-object action is constructed by parallel-processing, like multi-thread processing, among those primitive motions corresponding to all the dynamic patterns of its associated body parts. For an inter-object action like catch(), a given termination condition (e.g., collocated & hold, for catch()) and the kinematic constraints on its associated body parts (their length, degree of freedom, etc.) are used to functionally connect those body parts directly related to achieving its goal. That is, their motion patterns are determined not by its constituent primitive motions themselves as in the intra-object action, but by the sequence of primitive motions in a goal-directed manner. That sequence determines, under the kinematic constraints among functionally connected body parts, the trajectory from the current position of its associated uppermost joint to the goal point specified in its termination condition. When an agent interacts with an object or another agent in an inter-object action, their relative con-tact position varies according to contact timing. If a jogging agent tries to grab a water bottle from a table, for instance, his hand's exact position at his grabbing moment would vary along the continuum with respect to the bottle. Our approach can accommodate this variation in a real-time by dynamically retargeting the goal position according to the changing termination condition [11]. This dynamism cannot be realized with one action (method) using an interpolation technique based on key frames, where at least two sequential control points need to be prespecified for the interpolation.

As for the specific implementation schemes, our animation technique chooses a relative spatial coordinate system for agent modeling, which provides an effective basis for automatic satisfaction of kinematic constraints in reference to the uppermost (or root) body part as the base. Another scheme is direct transition to a new action independent of the previous action to implement real-time reaction to abrupt changes in situation. This scheme is generalized to an event level so that a new event occurs (or not) depending only on the current conditions of relevance, which enables dynamic event coupling. Also, interpolation in our technique is computationally performed using kinematics with respect to the terminal conditions of a motion instead of the Control Points [2]. The kinematics is used forwardly or inversely to generate intra-object and inter-object actions, respectively. In case of inter-object action, the termination condition is formulated in terms of a vector representing a relative inter-object position as in robotics, where only a relative position to a target object is relevant to exerting an effect on it [12]. This relative position vector along with kinematic chain (or hierarchy) allows our animation technique to maintain physical constraints and to reuse each (schematic) motion function in many different actions. The resulting animation is eventually bounded onto the global coordinates representing our entire cyber world, that is, the relative position vector computed via its associated kinematic equations is combined to the base vector of the root body link forming a global position and trajectory over time. Semantically this binding situates an agent's action in a global context, not as an isolated occurrence like in robotics [9], [10] but as a part of an intricate situation many interacting actions concur in. While an action of a robot is concerned only with its effect and associated kinetic links (or body parts), our simulation considers its procedure as significant as its effect and its entire body as well. No matter how minor a body part only an action involves, its animation scope is not confined to that functioning body part as in robotics [8] but comprehensive to include the entire body and its background situation. In fact an agent even in an idling state is meaningfully situated before being animated [13].

As a complex event in general can be successively decomposed into sub-events [14] and eventually animated in terms of iterated (or a continued) actions of agents, a realistic situation involving multiple events can be modeled and animated in different levels of inter-agent interaction and event coupling and their corresponding animation. Our animation technique enables an intricate multi-event situation to be depicted in as fine a level as intra-object motion on each agent involved, often via a concurrent or parallel action. All our schemes combined, the overall advantages of our technique over the conventional static (and monolithic) authoring [3], [15] encompass a fine-level agent interaction and event coupling, and adaptability to unpredictable development of a situation [16] along with motion variability and action diversity, and authoring scalability. These advantages collectively fulfill the objective in our simulation of realistic multi-event (and multi-agent) situations albeit at an expense of a limited visual realism. We implement several composite and parallel actions, and an intricate situation involving several concurring (some coincidentally coupled) events so as to demonstrate the viability of our animation technique.

The rest of this paper is organized as follows. Section 2 introduces related research works. Section 3 models agent's motions in different levels and actions. Section 4 introduces how each of those independent events concurring in a situation is decomposed into actions through event planning. Section 5 analyzes scalability and performance of our animation technique in comparison with conventional monolithic authoring and static animation. Section 6 describes the implementation. Section 7 draws a conclusion.

2. RELATED RESEARCH

While motion capture techniques provides visual realism [2], [3], their production costs would grow rapidly to record numerous scenes in application areas requiring user interaction or narrative variations. To improve reusability of captured data, Interpolation techniques analyze key frames to find similarity and interpolate body part trajectory via Knots in a sequence of key frames to produce animation [2], [3]. While the main advantage of Interpolation techniques is the low computational cost, the drawback is its requirement for the general assumption that the transition between parameters is linear or cubic and continuous in time, which is not always true. Though the Interpolation techniques may be similar to our agent modeling method in the general sense of reusing data, they unlike our method cannot abstract animation data into parameterized functions to be applicable to a variety of situations since they obtain key frames (or Knots) from 'monolithic' motion capture data [4], [5]. Rather than naming motions or actions through analysis of data pattern [3], our approach generates patterns of motions or actions by those functions formulating functional dependencies among body parts. That is, such a function generates motions and actions to fit diverse situations not by specifying body part positions like Knots in terms of points in motion capture data, but by parameterizing critical elements (position, angle, etc.) of motions or actions based on anthropomorphic kinematics similar to in robotics [10], [12].

In most interactive narratives, actions are uninterruptible or atomic [17], and their sequential order with respect to their entireties is the only way they are related to each other. Coupling between actions is rationalized by parameterizing behavior tree [13], mainly for code reuse in the context of a single event rather than interplay between independent events. In reference to the simplifying assumptions [17], our planning can have several advantageous characteristics for precise simulation of events, for example, the atomic time assumption is lifted, i.e., concurrent execution of actions is allowed; execution of an action is rendered interruptible; and its intermediate states are made visible and of concern to the planning agent.

While a narrative or robotics world usually is simplified in an abstract form or minimized in the form of spatial configuration geared to serving as stage or environment for a particular event or task in a small domain [10], [18], [19], an event or an action in our model is not intended to be an isolated occurrence but a part of an intricate situation in a real-world context. This contextualization requires a full-blown virtual background world represented in spatio-temporal coordinates, a sophisticated version of Working Memory [20], [21]. This full-blown background world provides our simulated situation with a historical context in which many contemporary events (or stories) are intertwined and their associated agents meaningfully interact with each other.

3. MODELING OF AGENT'S REUSABLE BEHAVIOR

We model an agent's body structure in terms of its body parts and their associated joints. We organize the body parts in layers based on their associated *kinematics equations* in order to systematically express its *anthropomorphic composition* and to implement the diversity of its motions at varying levels [22].

The resulting kinetic body structure can be abstracted in a pure tree structure based on joints connecting body parts [1], [12], [22], [23]. We naturally choose the pelvis as the root body part since it functions as a pivot in many composite motions and poses like 'walk' and 'sit' [22].

To formulate force propagation across the body parts, we use the world coordinate system instead of the model coordinate system, in that our virtual-world situations involve many objects mutually colliding over a wide space [24].

3.1 Modelling of Primitive Motions

Among the three levels of body movement, i.e., the primitive motion, composite motion, and action, primitive and composite motions each refer to the lowest level involving only a single joint, and that involving two or more joints. A set of the primitive motions is predefined for each joint, and some of them from different joints can be assembled in parallel to constitute a composite motion. Those composite motions may be arranged into an action. An action refers to a sequence of primitive and composite motions planned to achieve a specific goal.

To effectively visualize diverse motions, our animation method expresses the dynamic movement by changing the joint angle by setting a key frame as the goal state the moment the agent initiates a motion [2], [6], [7], [22]. The effect of a motion is decided according to a goal state assigned to the function for moving its associated joint. Joints are adjusted by their angles in an anthropomorphic way, that is, initial coarse joint movement followed by fine-tuning according to continuous visual observation. These functions are executed independently of each other for achieving real-time performance.

3.2 Modelling of Actions

Those primitive motions defined on each joint and composite motions can be arranged into an action. That is, given motion $m_K(t) = \prod_{i=1}^{M} \Theta_i(t)$, where $\Theta_i(t)$ denotes joint i's angle at t, action $a() = \prod_{t=0}^{N\cdot\Delta T} m_k(t)$ for continued type of action and $\sum_{i=1}^{N} m_K(i)$ for iterated type of action. Our hierarchical model of body structure allows for step-wise elaboration of motions or actions.

An action consists of n phases, $a = < p_1 \rightarrow p_2 \rightarrow p_3 \rightarrow \cdots \rightarrow p_n >$. With N motions $m_j, j = 1,2, \dots N$, comprising an action, a phase $p_i = \prod_j^N m_i$. A state S changes as a result of executing the motions m_j in an action [22].

$$S(t + \Delta t) = S(t) + \prod_{i}^{M} \left(\sum_{j}^{N} m_j^i, such\ as\ \mathcal{A}_s \supset m_j^i \right) \quad (1)$$

where S denotes a state, $\mathcal{m}$ denotes a motion, $\mathcal{A}$ denotes an action, M denotes the number of phases and N denotes the number of motions in each phase, $\prod$ denotes a phasic development and Σ denotes a set of parallel motions.

Multiple joints involved in a composite motion are to be properly synchronized and the possible difference in required degrees of precision among different body movements is to be resolved [20]. Real human's motions involve not only a pivot for a major rotation but other pivots for auxiliary rotations [25]. Because a primitive motion can only proceed in one direction, the number of phases in an action is to be increased according to the number of all the directions that changes can happen in the action [22], [26].

3.3 A parallel action by combining independent actions

An agent like a human in reality usually performs more than an action, as eating while walking. A parallel action refers to a composite action combining two or more actions performed by an agent at the same time. It contrasts with a concurrent action such as eating and talking, in that it is genuinely simultaneous, not ostensibly so. Consequently, it would require two or more body parts of the agent. If we have to treat each parallel action as a whole individually their animation would be intractable. We implement a parallel action by overlapping actions, which each are constructed in terms of motions [22]. These actions when combined into a parallel action may require their respective motions to be simultaneously executed by the same body part. However, multiple motions in general cannot be simultaneously executed by a body part, so their associated actions are prioritized according to how essential that body part is for their execution. We use three priority levels, namely 'essential', 'optional' and 'null', each indicating whether that body part is essential, optional or irrelevant (at least visually) for executing an action. Based on this priority, we evaluate the exclusivity between two actions on each body part using a priority chart [22]. They can be combined into a parallel action only if none of the body parts is exclusively needed by either of them. A parallel action combining two actions is implemented by animating in parallel disjoint motions between two composite motions corresponding to those actions.

Movements of an agent can be combined into an action module in any level, that is, primitive motions, composite motions, or actions, and that action module is to have a meaningful function. This action module parameterizes all the body parts that depend on the highest of those body parts essential to that action. Those body parts are all dependent directly or indirectly on that body part $\mathcal{P}_\mathrm{n}$ in the kinematic hierarchy, and additionally some of them depend on other parts $\mathcal{P}_\mathrm{i}$, i > n, as well, that is,

$$\mathcal{P}_\mathrm{n} > \mathcal{P}_\mathrm{i}, i = n+1, n+2, \dots\ k, and\ \mathcal{P}_\mathrm{i} > \mathcal{P}_\mathrm{j} \text{ for some i, j} > n \quad (2)$$

The number of component body parts, k, determines how precisely movement can be modeled.

3.4 Interaction between objects via inter-object actions

Interactions between objects are usually preceded by their spatial contact or collision [27]. When an action involves physical (usually visible) interaction between two objects or agents, its termination condition is further constrained by each other's conditions. Given their locations, $\vec{A}$ and $\vec{B}$, the distance, $\overrightarrow{A} - \vec{B}$, the visually appreciable aspect of an interactive action $a()$, determines N such that $\sum_{i=1}^{N} d(i) = \overrightarrow{A} - \vec{B}$, where the i-th primitive motion moves a distance, $\overrightarrow{d(\imath)} \leftarrow m_k(i \cdot \Delta T)$ with ΔT time increment. In case $\vec{A}$ or $\vec{B}$ is variable in time, i.e., $\overrightarrow{A(t)}$ or $\overrightarrow{B(t)}$, little additional animation cost is incurred except that for periodically updating that pair of position vectors because our assembly-based animation method uses motion functions flexible enough to handle quantitative situational variations. Of course the animation realism of a parallel action of constituent actions is proportional to how fine-grained their superposition is pre-formulated and animated according to their exact coinciding timings.

We classify the objects in our cyber world into the active objects and the passive objects. An active object like a human agent not only behaves autonomously but induces reactions from other objects. The use of priority queue in our agent model is to be paired with a message passing in a somewhat different sense from the conventional message passing, which is mainly for object encapsulation or distribution [28]. Not just the active objects of agents, those passive objects like a window are designed to perform functions like 'open' on receiving a message from an active object. This uniform message passing scheme is useful for visualizing a wide range of interactions beyond basic visual synchronization, e.g., between an agent and a bag it is carrying, That is, it can be used for implementing visual 'coherency', e.g., between two agents playing tennis, or between a customer agent placing an order and a clerk agent.

3.5 Uniform animation by reusing a small set of primitive motions

To animate the actions of all the objects in a situation requires a multi-tasking control mechanism. Even though all the motions on a scene look independent of each other, they technically belong to a single task requiring no context switching. A state of our cyber world resulting from the effect of many actions could be formulated by extending (1) to

$$S_k = \sum_{i}^{n} P_i \left(\prod_{q}^{n} \left(\sum_{j}^{n} f(g_j^q) \right) \right) + S_{k-1} \ (3)$$

where P_i denotes the agents, and the other notations are referred to (1).

A situation in our cyber world progresses through the set of phases in the actions all the agents perform. Agent action set $\{P_\mathrm{i}()\}$ consists of a number of $q's$ or sequentially planned steps of each action. Each step comprises motion functions that are gradually executed to achieve goal state g_j. When all those motion functions have reached their goal states in each step, the function $f(g_j)$ is initiated for their next goal states in a sequential order. That is, each action in an agent's plan can be animated based solely on those primitive motions independently generated. This animation method based on the primitive motions allows the actions to be combined or exchanged to react in real-time to changes in situation. Especially when an agent is involved in multiple events or faced by an abrupt change in situation, its reactive movement can be promptly animated in the form of parallel action or switching to a new action. We could model states just as realistic as they can be recognized so we could avoid using a physics engine to detect collisions like in a computation-heavy game.

4. ANIMATION OF COMPLEX INTER-EVENT SITUATION BASED ON REAL-TIME REACTION AND ACTION PARALLELISM

Our animation technique is designed to be meaningfully applied to agents acting in a dynamic situation involving unforeseen events along independent storylines, beyond isolated occurrence of movement in robotics or pre-authored single storyline in narrative systems. A real-world situation develops with a number of concurring events that are independently planned and executed by many autonomous agents. Some of those independent events would unexpectedly affect each other. This multitude of events and their coincidental coupling, though contributing to situation variability, would make prohibitively hard the situation simulation with conventional static intra-event planning and monolithic authoring methods [14], [16], [29]-[32]. We attempt to achieve both the situation variability and authoring scalability by a dynamic coupling of events. This dynamic event coupling is to be realized not only in the event planning and execution level but to be supported by diverse schemes in the motion animation level, which are coherently integrated into a novel real-time simulation paradigm. In the event level, a number of (schematic) events are pre-authored as a part of each agent's knowledge to be sequenced in its planning into an 'articulated' thread of events. All those schematic events are to be cast in a reusable format to be applicable to as many variations as possible. A schematic event refers to an event function parameterized in terms of variables corresponding to background-world conditions. In the execution phase, many such (branched) sequences of events, each corresponding to an independent agent's plan, are concurrently executed, so that it is likely for events from different sequences to be coincidentally coupled with each other. This coincidental inter-event coupling is made indirectly via some background-world factor that happens to be among their common parameters [20]. This coincidental coupling might create emergent situations by modifying events preexisting in the original plan or creating whole-new events. To realize this dynamic coupling, we need, as the event-level requirements, a full-blown background world for providing junction of connection, and association rules for directing the inter-event coupling as a premise for the inter-event planning and execution, and, as the action-level requirements, real-time reaction to unforeseen situations and efficient handling two or more simultaneous actions in independent events as well as effective animation of diverse actions. Notice our dynamic coupling via real-world rules conditioned on the real-time state of the background world also provides a foundation for resolving the authoring scalability. That is, the relationships or impacts among the parameters of events are automatically reflected on the development of events and inter-event connections since all those parameters are semantically interconnected via their common background world.

We briefly describe how a number of pre-planned and unforeseen events alike can be dynamically coupled into a coherent situation, possibly ramifying into many storylines. All those anticipated events are derived initially in search-based planning and then possibly combined with unforeseen events during their execution. Their actual coupling is designed to occur in the fine level of agents' actions which constitute the basic units of animation in our simulation. This intimate and coincidental coupling is implemented in action level with respect to: interleaved arrangement among 'exogenous' groups of actions in each agent's plan, dynamic scheduling in reaction to incessantly-changing background world conditions, and parallel action for playing a multitude of concurring roles in different events. We elaborate along an example situation how these techniques are integrated for simulating a realistic situation involving several independent events and autonomous agents. The following example situation comprises independent story-lines such as a ball play, a party and a (conditional) medical treatment, each involving many events which could be coincidentally coupled to each other.

4.1 Schematic event planning for a goal

Given a goal, an agent identifies an event believed to help achieve that goal. Such an (over-arching) event in general is complex enough to demand a deliberate planning in terms of smaller events, eventually of animated actions performed by its agent(s) [21]. Initially, a plan is drawn up based on a search of the planning agent's knowledge about the background world, where the planned event would be executed. Such a search would never be exhaustive but be as extensive as its associated agent could foresee, ramifying to several potential paths in a graph of events as illustrated in Figure 1. Those events (denoted by three-segment ovals) in each path are successively identified starting from the one able to immediately satisfy the goal according to the functional association (denoted by a chained pair of broken and solid arrows), in such a way that the effect of an event produces a part of precondition of another event via their common background factors (denoted by dark-green circles.) The associations between situations and events provide a key to logical deduction of relevant events in planning for an overarching event involving many sub-events [21]. For instance, the 'party' event would be chained backward with 'guest invitation', 'cooking food', 'arranging place', etc. The actions resulting from decomposition of those identified events are instantiated along the timeline to constitute a schedule, which, if executed accordingly, could accomplish the goal. That is, those actions are arranged into a branched sequence, each action corresponding to a leaf node of its associated hierarchical event plan [33].

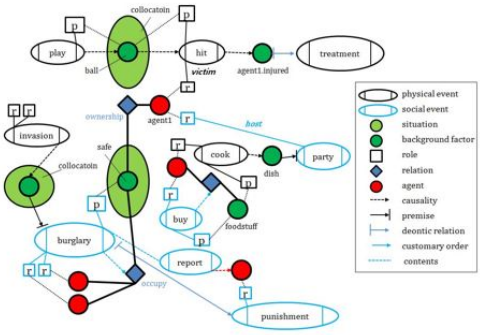


Fig. 1. Multiple threads of reasoning for planning in multi-event situation

4.2 Action-level execution mechanism of multi-event situation

We briefly present a (priority) queue-based execution mechanism by which independent events in a situation are naturally coupled with each other in the fine level of actions. As a result of the initial (schematic) planning, each overarching event identified through the search has been decomposed into actions. All those identified actions are interleaved in between the existing (action) elements of the priority queues corresponding to their associated agents' intentions. Those existing elements would originate from earlier plans involving those agents, independent of this present plan. While the intra-event precedence among sub-events (or actions) as identified in the event planning phase are to be strictly maintained across their respective queues, the inter-event priority among independent events may be observed in a sequential or an interleaved manner with respect to their respective sequences of actions. These initial priorities could be rearranged possibly to a new order if unforeseen (or exogenous) events are enqueued as a result of a coincidental event coupling in execution time. Despite of this rearrangement, the relative intra-event order among those actions belonging to each event would be preserved even if the interval between actions may vary due to an embedded exogenous event (or its component actions.)

An intra-agent coupling presents a clue for action-level parallelism between the action the agent currently performs and a future plan (or an unforeseen action) while an inter-agent coupling occurs in interaction among agents interrelated via the background world. A change in the overall plan due to a new event arising from change in the background world or a re-planning of an existing event is realized in the action unit. Since pre-planned events and unforeseen exogenous events are all in the same form of actions in a queue, they are uniformly executed in terms of actions scheduled on the same agent's timeline leading to possibility of being combined into a parallel or concurrent action on an agent [9], as with 'making a phone call walking to a shop', 'dodging a flying ball carrying a dish', etc. This dynamic reprioritizing of events and associated rearranging of their component actions allows independent events to be coincidentally integrated via the background world into a realistic emergent situation.

We developed a (priority) queue-based multi-event execution mechanism, which is basically organized in three layers, event, action and motion layers in a top-down order. The event layer arranges events, planned and unforeseen, in a situation according to their priority or urgency. Many sequences of actions corresponding to as many events are distributed into their associated agents' queues according to their interleaved order as scheduled in a global (a situation) level [21].

5. AUTHORING SCALABILITY AND REAL-TIME ANIMATION FOR MULTI-EVENT SITUATIONS

The conventional monolithic authoring as in IS [14], [16] would suffer from scalability problem especially when the variability of situations is desirable [34], [35] since that method needs to author motions or scenes *anew* for each of the different storylines or event flows, and to visualize each of the entire situations *anew* for any minor change in their background. A greater variability of story or a greater number of sub-events involved in an intricate (i.e., realistic) situation is translated to a greater authoring burden and its corresponding real-time animation requirements. To alleviate this scalability problem on the event-planning level, dependence of each sub-event of an event (or scene) on its previous sub-events is to be eliminated from the monolithic authoring, where all those sub-events are authored all in one cluster. That is, each sub-event of an event is to be *dynamically* coupled in its execution time according to the associated rules with other sub-events in reference only to the current conditions of the background world, and further to be coincidentally connected with other independent events to form a situation.

To quantify the burden of a monolithic authoring for an overarching event comprising n steps, we need to author at a minimum, $N_1 \cdot (k \cdot l \cdot m) \cdot N_2 \cdot (k \cdot l \cdot m) \cdot N_3 \cdot (k \cdot l \cdot m) \cdot \ldots \cdot N_n \cdot (k \cdot l \cdot m) = N_1 \cdot N_2 \cdot N_3 \cdot \ldots \cdot N_n \cdot (k \cdot l \cdot m)^n$, where N_i denote the number of candidate sub-events at the i-th step; and k, l and m denote the average number of instances per situation, the average number of attributes per situation, and the average number of domain values per attribute, respectively. In contrast, our authoring is based on dynamic event and action coupling which allows each step to be animated independent of its previous steps. As a result, we need to author a (potential) sub-event no more than once even if a sub-event ramifies into different sub-events on a storyline to form as many situation variations. That is, the authoring burden in terms of the number of events could be estimated as: $N_1 \cdot (k \cdot l \cdot m) + N_2 \cdot (k \cdot l \cdot m) + N_3 \cdot (k \cdot l \cdot m) + \ldots + N_n \cdot (k \cdot l \cdot m) = (N_1 + N_2 + N_3 + \ldots + N_n) \cdot (k \cdot l \cdot m)$, which is a sharp contrast to $N_1 \cdot N_2 \cdot N_3 \cdot \ldots \cdot N_n \cdot (k \cdot l \cdot m)^n$. However, this dynamic (or coincidental) coupling naturally is to be supported by a real-time animation technique.

The authoring scalability in motion (or action) level could be estimated in a similar manner as in event (or situation) level. Since reusing motion capture data in a conventional key frame technique may be possible over a part of frame sequence but not with a part of image on a frame [3], any partial change in an image pattern on a frame would require recapturing of the entire frame (sequence.) That is, if an agent is modeled as a kinematic chain with m joints and joint k_i is capable of performing n_i kinds of movements, the maximum number, N, of motion types the agent can perform would be, $\mathrm{N} =$

$(n_1 \times n_2 \times \ldots \times n_m)$. In reality, a practical sub-set of motions, N', could be far smaller due to anthropomorphic constraints, i.e., $N' \ll N$. To animate even these N motions, our assembly-based reusing technique requires to author only $n_1 + n_2 + \cdots + n_m$ motion functions whereas a motion capture technique or conventional key frame technique would need to create $n_1 \times n_2 \times \ldots \times n_m$ sequences of motion images. To sum up, the overall advantage in authoring scalability of our approach based on dynamic coupling of events and corresponding real-time animation schemes over the conventional technique is significant, and grows exponentially as the situations (or variations in each step of an event) multiply in our simulation. Admittedly an additional cost is required for authoring schematic actions in a flexible form, and association rules to be used for their coupling.

Further, this dynamic (or coincidental) coupling naturally is to be supported by a real-time animation technique for handling diversity and variability of actions stemming from the intricacy and non-determinism of situation involving a number of independent events, and for promptly reacting to abrupt changes in the scenes [14] due to coincidental event coupling. This real-time animation has been realized in our approach using the parallel actions and reusability of actions (and motions) together with prompt reaction capability provided in our approach as described earlier.

6. IMPLEMENTATION OF EXAMPLE ACTIONS AND MULTI-EVENT SITUATIONS

We implemented our animation technique that is based on the reuse of motions and superimposition of composite motions. We demonstrate the viability of our method with respect to real-time reaction and action parallelism on top of motion reusability. Of those implemented actions such as 'walk', 'open', 'handshaking', 'eating while walking' and 'dodging','walk' is an action implemented as an iterated composite motion; 'open' is an inter-object action involving an interaction between an agent and a door (object) ; 'eating while walking' is a parallel action between 'eat' and 'walk'; two actions are reactively animated in response to unforeseen events, one of those actions is the reflexive and parallel type as well. We show how many of these (schematic) actions are reused to animate a multi-event situation involving independent events, which are coincidentally coupled in execution time. Our implementation tools include Visual studio 2008 and OpenGL, along with the Microsoft Access 2007 as the database environment.

The sequence of scenes S1 through S6 in Figure 2 shows an agent performing a movement toward a given goal location by executing 'walk' motion iteratively. This composite motion is animated by assembling several primitive motions into a pose and activating that pose according to its associated kinematic equations. The primitive motions of each arm and leg are synchronized in 'walk' motion by using different motion speeds to fit their respective travelling distances. The (schematic) action function is instantiated with its parameter values corresponding to the initial and destination locations and executed as many times as needed until reaching that destination. The destination may be inserted by the user or internally generated by the agent to be used to instantiate a parameter of that schematic action function.

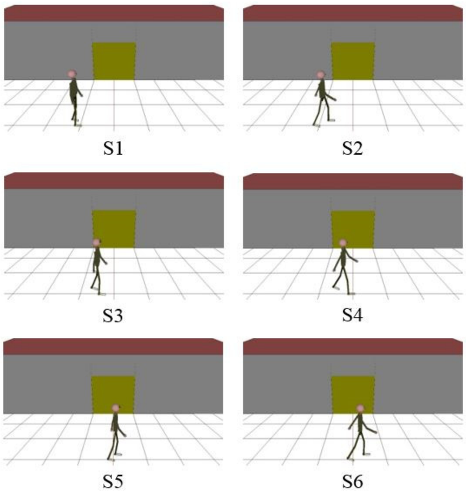


Fig. 2. A sequence of motions for composite action of 'walking'

To perform an inter-object action, an active object is designed to interact with other objects by sending messages to them. The messaging frequency is adjusted to be proportional to the occurrence frequency of unanticipated (noticeable) changes, which allows for exact depiction of interaction and synchronization in incremental details as needed, and consequently abrupt (unusual) change. Our technique exploits this uniform messaging scheme across the object types to extend the interaction types to social interactions beyond ones caused by physical contacts as explained earlier. The following sequence of scenes S1 through S4 shown in Figure 3 animates an agent opening a door. The active agent object and the passive door object exchange messages for their synchronization.

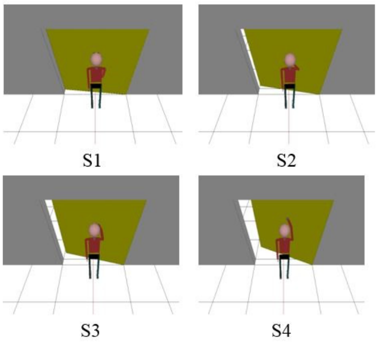


Fig. 3. A Sequence of motions for inter-object action of 'opening a door'

In a real situation, it is an important capability for an agent to promptly react to an abrupt change in ambient state. Figure 4 shows a situation where two agents coincidentally interact each other. While Agent 1 is walking from location L1 to location L2 according to its original plan, it runs into a friend, Agent2, (say, at an intermediate location L3 on its way) in Scene S2. In its reaction, Agent 1 would greet him with a handshake before resuming (after an instant re-planning) its moving event to L2. This unforeseen event would require to be deliberately pre-authored separately from the original event in terms of branched sequences of animated scenes in conventional monolithic authoring [14].

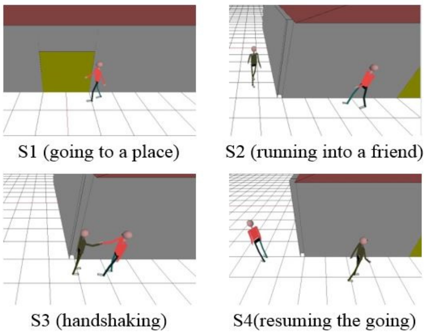


Fig. 4. A Sequence of motions for 'walking' and coincidental 'handshaking'

In contrast, our technique implements such a coincidental interaction in an action (or motion) level without pre-authoring of different events. That is, a 'walk' action (occurrence) is switched to a 'handshake' action (occurrence) in execution time conditional on its associated ambient state. Notice these two actions could even be combined into a parallel action if appropriate, e.g., Agent 2 is in a hurry, furthering the intimacy of event coupling.

Figure 5 shows, in a series of scenes S1 through S6, an agent walking while eating. This parallel (or concurrent) action like a new composite action is effectively constructed using pre-existing actions (or a sequence of motions) rather than using primitive motions. That is, an 'eating' action is superimposed on top of a 'walking' action instead of assembling all the primitive motions like 'leg-lift()', 'bite()', etc. The superimposion pattern is determined by the priority table described earlier. This parallelism is applicable to any two actions that each exclusively use only their respective disjoint sets of body parts. For example, 'eat' exclusively uses mouth and hand, while 'walk' uses legs. The resulting scene is visually equivalent to an iterated composite motion.

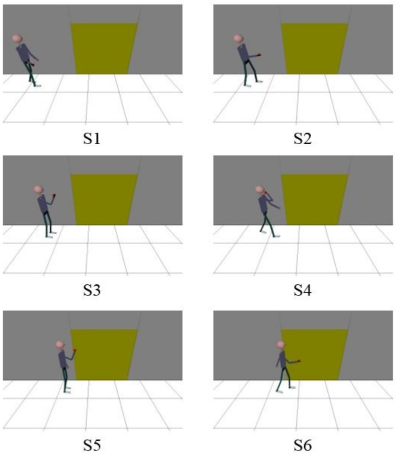


Fig. 5. A sequence of motions for parallel action of 'eating while walking'

When two independent events are abruptly coupled with each other, their coupling may occur as intimately as on one agent's body involved in both events. This intra-body coupling would require to visually create a parallel action, and that usually in no time. As shown in Figure 6, this example emergent situation culminating in S3 occurred as the cook carrying food for the party happened to be on a collision course with the ball flying from an independent ball-playing event. An original action of carry() and a reflexive action of dodge() in an instinctive reaction to an unforeseen peril can be superposed on their common agent, the cook, into a parallel action. Internally, dodge() is designed to be invoked immediately from a pre-fabricated motion function instead of from a usual action sequence derived by search-based planning in order to meet its real-time animation requirement for a prompt reaction. Notice this emergency situation involves all the three focal real-time animation issues stated earlier, that is, coincidental interaction between agent and object, each agent's handling of multiple roles, and prompt reaction to abrupt change of an ambient condition.

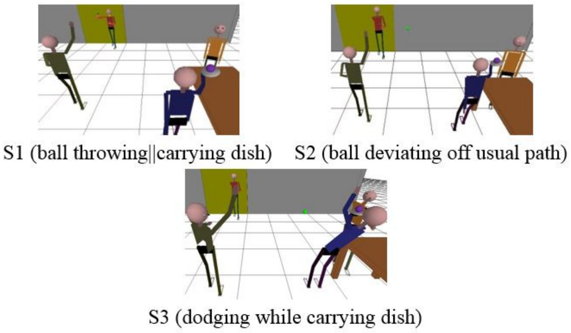
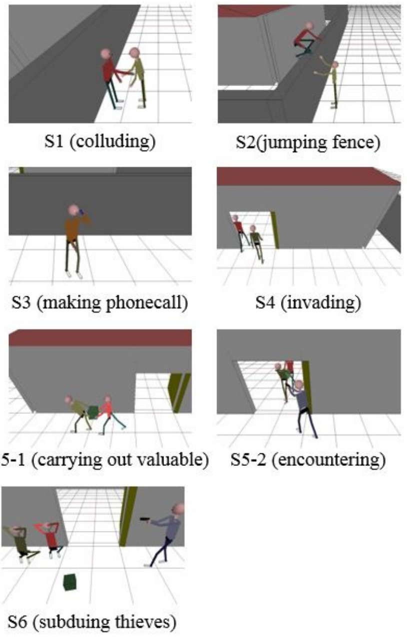

S1 (ball throwing||carrying dish)
S2 (ball deviating off usual path)
S3 (dodging while carrying dish)

Fig. 6. A Situation involving 'carry' event and reactive action of 'dodge'

While a complex event in general involves many agents and objects, its development could be animated in terms of the actions of its associated agents. Consequently, its scenes can be visually implemented in terms of the sequences of individual actions as we proposed. That is, the parallel or concurrent actions for individual agents and the planning scheme for achieving a goal mentioned earlier can be directly reused for a situation involving multiple agents. In Figure 7, several scenes feature four agents performing diverse inter-object actions involved in 'theft', 'report' and 'arrest' events. Those actions in scenes S1 through S6 are: 'the agents talking to each other', 'an agent helping the other jump the fence', 'an agent using a phone', 'the agents invading together', 'carrying out a heavy box together', 'the agent arresting the other two on their encounter', etc. Some action (occurrences) are tentative as between scenes S5-1 and S5-2, depending on their associated conditions, e.g., whether the policeman agent arrives at the crime scene in time or not, and which action sequence the situation will actually unfold along can be only non-deterministically decided.



S1 (colluding)

S2(jumping fence)

S3 (making phonecall)

S4 (invading)

S5-1 (carrying out valuable)

S5-2 (encountering)

S6 (subduing thieves)

Fig. 7. A sequence of inter-agent actions for event of 'theft'

Notice that all these actions can be visualized by reusing (in sequences) a small set of basic actions such as 'speak', 'gesture', 'walk', 'hold', 'pull' and 'climb'. Each individual inter-object (inter-agent) action is formulated to be adaptable to variations not only in their associated conditions in situation, but also in relative occurrence timings between their associated actions or relative positions of their associated objects. Using this assembly-based animation method with reusable motion and action functions, numerous interaction patterns of inter-agent action (occurrences) along a continuum of their relative occurrence timing can be animated without pre-authoring all those variations, as exemplified below in 'helping jump fence', 'carrying box' or 'shaking himself free from'.

7. CONCLUSION

The efficient implementation of diverse physical actions of agents to respond to dynamically changing situations is essential for the simulation of realistic agents and activities in a cyber-world. To achieve a maximum diversity of actions and immediate responsiveness to abrupt changes in a situation, we have developed an animation technique in which complex actions are recursively constructed by reusing a set of primitive motions, and agents are designed to react in real-time to abrupt ambient changes by computationally satisfying kinematic constraints on body parts with respect to their goals.

We first model the agent's body as a kinematic hierarchy with its body parts functionally interconnected via joints, and its movement is modeled by the kinematic equations of the hierarchy. Based on this kinematic model, we defined a set of primitive motion in terms of parameterized motion functions, and they are sequenced successively into more complex motion functions and diverse action functions. This assembly-based animation scheme also allows multiple actions simultaneously performed by an agent to be naturally combined into a parallel action. Computational generation of motions under their associated kinematic constraints allows key frames to be reflected on the motion function during the animation process for a body part, which constitutes an essential real-time animation scheme for simulating agents' prompt reactions in an emergent situation involving coincidental inter-agent interaction. Unlike an intra-agent action, an inter-agent action is further elaborated in terms of contact timing between its associated agents in order to enhance situation variability. These real-time animation schemes for motion variability and action diversity along with prompt responsiveness are applied to animating a multi-event situation, where independent events (often their associated agents) tend to be coincidentally coupled with each other [14]. We presented an action-level execution mechanism to elaborate on how a coincidental coupling between concurrent events can be animated, possibly in as intimate a degree as intra-agent action, that is, parallel action.

Our implementation schemes include using a relative spatial coordinate system for agent modeling as an effective basis for automatic satisfaction of kinematic constraints, direct transition to a new action independent of the previous action to realize real-time reaction, computational interpolation using kinematics with respect to the terminal conditions of a motion, and agent interaction and event coupling in the intra-object action level. All these schemes combined, our animation technique accomplishes those overall advantages, over the conventional static (and monolithic) authoring-based narrative systems, such as motion variability and action diversity, and authoring scalability, a fine-level agent interaction and event coupling, and adaptability to unpredictable development of a situation. Above all, a real-time animation is achieved along with high motion variability and action diversity. We analytically quantified our advantage in the authoring scalability. These advantages collectively fulfill the objective in our simulation of realistic multi-event (and multi-agent) situations albeit at an expense of a somewhat limited visual realism. We demonstrated the viability of our animation technique by implementing several composite and parallel actions in an intricate situation involving several concurring (some coincidentally coupled) events.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(Ministry of Education) (No. 2016R1D1A2B03935650).

REFERENCES

[1] J. Lee, *Modeling Human Actions based on Primitive Action and Class Inheritance*, The Graduate School Kyungpook National University, 2009.

[2] G. Xia, H. Sun, X. Niu, G. Zhang, and L. Feng, "Keyframe extraction for human motion capture data based on joint kernel sparse representation," IEEE

Transactions on Industrial Electronics, vol. 64, no. 2, 2017, pp. 1589-1599. doi : 10.1109/TIE.2016.2610946

[3] Daniel Lewkowicz and Yvonne Delevoye-Turrell, "Real-Time Motion Capture Toolbox (RTMocap): an open-source code for recording 3D motion kinematics to study action–effect anticipations during motor and social interactions," Behavior research methods, vol. 48, no. 1, 2016, pp. 366-380. doi : 10.3758/s13428-015-0580-5

[4] Richard Roberts, J. P. Lewis, K. Anjyo, J. Seo, and Y. Seol, "Optimal and interactive keyframe selection for motion capture," In SIGGRAPH Asia 2018 Technical Briefs, 2018, p. 26.

[5] William Li, Lyn Bartram, and Philippe Pasquier, "Techniques and Approaches in Static Visualization of Motion Capture Data," In Proceedings of the 3rd International Symposium on Movement and Computing, 2016, p. 14.

[6] S. Ryu, Y. Park, B. Kyung, D. Lee, and W. Lee, "3D game character animation pipe-line to improve utilization of motion capture," Journal of Korea Contents Association, vol. 8, no. 7, 2008, pp. 120-127, doi : 10.5392/JKCA.2008.8.7.120

[7] T. Choi, *Abstract of digital motion capture system and 3D game character animation application*, Kongju National University, 2006.

[8] R. Parent, *Computer Animation - Algorithms and Techniques*, Ajin publishing company, 2004.

[9] Guillermo Díaz and Andrés Iglesias, "Swarm Intelligence Scheme for Pathfinding and Action Planning of Non-player Characters on a Last-Generation Video Game," International Conference on Harmony Search Algorithm, Springer, Singapore, 2017.

[10] O. Khatib, E. Demircan, V. De Sapio, L. Sentis, T. Besier, and S. Delp, "Robotics-based synthesis of human motion," Journal of Physiology-Paris, vol. 103, 2009, pp. 211-219. doi: https://doi.org/10.1016/j.jphysparis.2009.08.004

[11] Du Sel, Yann Pinczon, Nicolas Chaverou, and Michaël Rouillé, "Motion retargeting for crowd simulation," Proceedings of the ACM Symposium on Digital Production, 2015, pp. 9-14.

[12] Jens Kober, Matthew Glisson, and M. Michael, "Playing catch and juggling with a humanoid robot," Humanoid Robots, 12th IEEE-RAS International Conference on IEEE, 2012.

[13] A. Shoulson, F. M. Garcia, M. Jones, R. Mead, and N. I. Badler, "Parameterizing Behavior Trees," Motion In Games, 2011, pp. 144-155. doi: https://doi.org/10.1007/978-3-642-25090-3_13

[14] F. Charles, M. Lozano, S. Mead, A. F. Bisquerra, and M. Cavazza, "Planning formalisms and authoring in interactive storytelling," Proc. of TIDSE, vol. 3, 2003.

[15] J. Porteous, M. O. Cavazza, and F. Charles, "Applying planning to interactive storytelling: Narrative control using state constraints," ACM Trans. on Intelligent Systems and Technology, vol. 1, no. 2, Nov. 2010. doi: 10.1145/1869397.1869399

[16] M. Cavazza, F. Charles, and S. J. Mead, "Emergent situations in interactive storytelling," Proc. of ACM Symposium on Applied Computing (ACM-SAC), Madrid, Spain, 2002.

[17] O. Mark Riedl and Vadim Bulitko, "Interactive narrative: An intelligent systems approach," Ai Magazine, vol. 34, no. 1, 2012, p. 67. doi: https://doi.org/10.1609/aimag.v34i1.2449

[18] K. Hartsook, A. Zook, S. Das, and M. Riedl, "Toward supporting stories with procedurally generated game worlds," Proc. of the IEEE Conference on Computational Intelligence in Games, 2011.

[19] A. Shoulson, M. Gilbert, M. Kapadia, and N. Badler, "An Event-Centric Planning Approach for Dynamic Real-Time Narrative," Proc. MIG'13, Dublin, Ireland, 2013.

[20] P. Winston, *Artificial Intelligence*, Addison Wesley Publishing Co., 1992, pp. 119-161.

[21] J. Park, "Implementation of an agent-centric planning of complex events as objects of pedagogical experiences in virtual world," Int'l Journal of Contents, vol. 12, no. 1, Mar. 2016, pp. 25-43, doi: http://dx.doi.org/10.5392/IJoC.2016.12.1.025

[22] J. Choi and J. Park, "An Effective implementation of agent's complex actions by reusing primitive motions," Proc. of Simultech, Vienna, Austria, Aug. 2014.

[23] S. Pitzel, *Character Animation: Skeletons and Inverse Kinematics*, https://software.intel.com/en-us/articles/character-animation-skeletons-and-inverse-kinematics, 2017.

[24] R. Wright, T. Lipchak, H. Choi, *OpenGL super bible,* Information Publishing Group, 2005, pp. 190-219.

[25] J. Diebel, *Representing Attitude: Euler Angles, Unit Quaternions, and Rotation Vectors*, Stanford University, 2006.

[26] David Tlalolini, Christine Chevallereau, and Yannick Aoustin, "Human-like walking: Optimal motion of a bipedal robot with toe-rotation motion," IEEE/ASME Transactions on Mechatronics, 2011, pp. 310-320.

[27] P. Winston, *Artificial Intelligence*, third edition, Addison Wesley Publishing Co., 1992, pp. 323-346.

[28] T. Kim, "Representation of physical Phenomena and Spatial Relations in the virtual reality," Journal of Korea Contents Association, vol. 12, no. 6, 2012, pp. 21-31, doi: https://doi.org/10.5392/JKCA.2012.12.06.021

[29] A. Goldberg and D. Robson, *Smalltalk-80 The Language*, Addison Wesley, 1989, pp. 5-16.

[30] B. Magerko, "Evaluating preemptive story direction in the interactive drama architecture," Journal of Game Development, vol. 2, no. 3, May. 2007, pp. 25-52.

[31] Aylett Ruth, Sandy Louchart, Anders Tychsen, Michael Hitchens, Rui Figueiredo, and Carlos Delgado-Mata, "Managing emergent character-based narrative," Proc. of the 2nd international conference on Intelligent Technologies for interactive entertainment. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[32] R. Aylett, M. Vala, P. Sequeira, and A. Paiva, "FearNot! - An Emergent Narrative Approach to Virtual Dramas for Anti-bullying Education," Proc. of International Conference on Virtual Storytelling, 2007, pp. 202-205.

[33] Rui Figueiredo, Antonio Brisson, Ruth Aylett, and Ana Paiva, "Emergent stories facilitated," Interactive

Storytelling, 2008, pp. 218-229. doi: https://doi.org/10.1007/978-3-540-89454-4_29

[34] Leslie Pack Kaelbling and Tomás Lozano-Pérez, "Hierarchical task and motion planning in the now," Robotics and Automation (ICRA) IEEE International Conference, 2011.

[35] R. Hodhod, P. Cairns, and D. Kudenko, "Innovative integrated architecture for educational games: challenges and merits," Trans. on edutainment v, 2011. pp. 1-34. doi : https://doi.org/10.1007/978-3-642-18452-9_1

Jong Hee Park
He acquired his Ph.D in Computer Engineering from Univ. of Florida in 1990. He has since been on the faculty at Kyungpook Nat'l Univ. in Korea. His research interests focuses on intelligent systems based on cyber-worlds. The specific research topics encompass intelligent tutoring system for immersed language learning, futuristic simulation games, and simulation of cyber-worlds for diverse applications.

Jun Seong Choi
He received his B.S. in Electronics Enginnering from Kyuongpook National University, Korea in 2014, Currently Ph.D course student, in Kyungpook National University, Korea. His research interest is Artificial Intelligence, specifically animation and planning techniques in multi-event.