Queueing network-adaptive control of thought rational (QN-ACTR): an integrated cognitive architecture for modelling complex cognitive and multi-task performance

Shi Cao* and Yili Liu

Department of Industrial and Operations Engineering,
University of Michigan,
Ann Arbor, 48109-2117, Michigan, USA
E-mail: shicao@umich.edu
E-mail: yililiu@umich.edu
*Corresponding author

Abstract: How to computationally model human performance in complex cognitive and multi-task scenarios has become an important yet challenging question for human performance modelling and simulation. This paper reports the work that develops an integrated cognitive architecture for this purpose. The resulting architecture – queueing network-adaptive control of thought rational (QN-ACTR) – is an integration of the QN mathematical architecture and the ACT-R symbolic architecture. This integration allows QN-ACTR to overcome the limitations in each method and model a wider range of tasks. Implemented as a computerised simulation programme, QN-ACTR has been verified in the simulation of 20 typical tasks from the ACT-R literature. The benefits of the integration have been demonstrated in the simulation of 29 transcription typing phenomena, showing its capability in modelling complex cognitive and multi-task scenarios that have not been modelled by either QN or ACT-R.

Keywords: queueing network; cognitive architecture; complex cognition; multi-task performance; human performance modelling; ACT-R; human factors simulation.

Reference to this paper should be made as follows: Cao, S. and Liu, Y. (2013) 'Queueing network-adaptive control of thought rational (QN-ACTR): an integrated cognitive architecture for modelling complex cognitive and multi-task performance', *Int. J. Human Factors Modelling and Simulation*, Vol. 4, No. 1, pp.63–86.

Biographical notes: Shi Cao is a PhD candidate within the Department of Industrial and Operations Engineering at the University of Michigan, Ann Arbor. He received his MEd in Psychology with focuses on human factors and cognitive modelling from Zhejiang University, Hangzhou, China, in 2007. His research areas include human performance and cognitive modelling, healthcare human factors, and transportation human factors.

Yili Liu is Arthur F. Thurnau Professor of Industrial and Operations Engineering at the University of Michigan. His research areas include computational models of human performance, cognitive engineering, and engineering aesthetics. He has published in the theoretical and applied journals and contributed to industrial projects.

This paper is a revised and expanded version of a paper entitled 'Integrating queueing network and ACT-R cognitive architectures' presented at the Human Factors and Ergonomics Society 55th Annual Meeting, Las Vegas, NV, September, 2011 and a paper entitled 'QN-ACTR modeling of multitask performance of dynamic and complex cognitive tasks' presented at the Human Factors and Ergonomics Society 56th Annual Meeting, Boston, MA, October 2012.

1 Introduction

The design and evaluation of human-machine systems need computational models to quantitatively simulate and predict system performance and reliability. This requirement, however, has become increasingly difficult to meet, as the complexity of human-machine systems grows rapidly. Common tasks in the workplace today often require multi-task performance and complex cognitive activities in addition to physical activities. For example, a typist often needs to type quickly on the keyboard and comprehend the texts, while responding to incoming short text messages. An automobile driver often needs to control the vehicle's lane position and speed while communicating on a mobile phone. A physician in an emergency department needs to monitor multiple vital signs of multiple patients and make critical decisions under time pressure. This increase of task complexity raises an important yet challenging human factors question: How to model and simulate human performance in complex cognitive and multi-task scenarios?

One answer to this question comes from developing integrated computational cognitive architectures. A cognitive architecture is a comprehensive representation of the human mind and possesses the following properties. First, it integrates isolated cognitive theories under the same framework where different mental modules execute different aspects of mental functions. Second, the modules and their mechanisms are biologically inspired, based on psychological and neurological evidence of how the human brain works. Third, a cognitive architecture can be implemented as a computerised simulation programme to generate simulated behaviours and quantitatively predict human performance. Since human performance involves all aspects of cognition including perception, memory, and response selection, a predictive model needs to unify all the underlying cognitive mechanisms (Newell, 1990, 1973). For instance, in a study that reviewed existing models related to helicopter pilot performance, researchers found hundreds of isolated models and concluded that the central problem is to integrate them into a coherent unity that works together (Elkind et al., 1989).

Over the past several decades, numerous efforts have been made in developing unified theories of cognition and led to the creation of several important cognitive architectures including Adaptive control of thought-rational (ACT-R; Anderson et al., 2004), Executive-process interactive control (EPIC; Meyer and Kieras, 1997), Soar (Laird et al., 1987), and Queueing network-model human processor (QN-MHP; Liu et al., 2006). The evolution of these architectures is a testimony to model integration. ACT-R originally focused on modelling pure cognitive activities and later acquired perceptual-motor modules conceptually based on EPIC's perceptual-motor processors

(Byrne and Anderson, 1998). Soar's declarative memory modelling mechanisms are inspired by ACT-R's declarative knowledge representation and computations (Laird, 2008). QN-MHP integrates mathematical queueing network (QN) modelling with model human processor (MHP) and goals, operators, methods, and selection (GOMS) rules models (Card et al., 1983; John and Kieras, 1996); QN modelling itself has integrated a wide range of mathematical models of mental processes (Liu, 1996, 1997). Although previous work of model integration has significantly expanded the scope of tasks that can be modelled by a cognitive architecture, existing architectures still have difficulties in modelling human performance in multi-task scenarios involving complex cognitive activities. ACT-R and Soar have the capability to model complex cognition such as reading comprehension, learning, and decision making, but they lack a mechanism that can schedule multi-task processing at the local module level. EPIC focuses on modelling perceptual-motor tasks but lacks the mechanisms to model complex cognitive activities that involve declarative memory, for example, language learning and sentence comprehension. QN-MHP uses queueing as a multi-task coordination mechanism but is limited in modelling complex cognition. These limitations indicate the need for further model integration.

The present study continues along the line of work using QN to model and simulation human performance (Liu, 1996, 2009; Liu et al., 2006; Wu and Liu, 2008a). Although QN and queueing theory are well-established mathematical formulations that have been widely used in the modelling and simulation of complex engineering systems such as telecommunication and factory layout, they were not explicitly used in the modelling and simulation of human mental performance until recently. Nevertheless, there have been evidences that suggest the existence of queues in the cognitive system. At the neuron and synapse level, it has been found that the mobilisation of synaptic vesicles (which store and release neurotransmitters to transfer information between neurons) follows a queueing mechanism that gives some vesicles higher priority than others (Holt and Jahn, 2004). At the cortex level, motor commands are also believed to form queues in the human brain before their execution.

> "A sequence of different isolated finger movements requires programming in the supplementary motor areas. We suggest that the supplementary motor areas are programming areas for motor subroutines and that these areas form a queue of time-ordered motor commands before voluntary movements are executed by way of the primary motor area". [Roland et al., (1980), p.118]

As discussed by Liu (2009), the further integration of the QN mathematical architecture and other symbolic architectures can examine and potentially resolve several theoretical and methodological issues in human performance modelling. In particular, the integration of QN and ACT-R architectures has the potential to model performance and mental workload in multi-tasks with complex cognitive components (Cao and Liu, 2011a, 2011b, 2012b). Continuing model integration and the development of a unified theory of cognition, the goals of the present study are to integrate QN and ACT-R by merging their structures and modelling mechanisms and build a cognitive architecture that can model a wider range of cognitive and concurrent tasks. This integration also allows the examination of several important theoretical issues in ACT-R – including concurrent goal scheduling and module jamming – from the QN perspective.

Modelling complex cognitive and multi-task scenarios

The integration of QN and ACT-R combines QN's strength in modelling multi-task performance and ACT-R's strength in modelling complex cognition. Two unique theoretical positions in QN are the queueing mechanism and the hybrid server network (Liu et al., 2006; Wu and Liu, 2008a). The queueing mechanism can serve as a natural multi-task coordination mechanism at the local server level, and the hybrid network with both serial and parallel processing servers can model multi-task processing with a finer granularity. Previous studies using the QN architecture (i.e., QN-MHP) have modelled multi-task scenarios including the psychological refractory period dual-tasks (Wu and Liu, 2008a) and a driving and map viewing dual-task (Liu et al., 2006). On the other hand, ACT-R's strength lies in its symbolic knowledge representations and sub-symbolic computations that can model memory retrieval and learning. ACT-R assumes two types of knowledge representations: chunks and production rules (i.e., rules, for short). Chunks represent the declarative knowledge of facts and indicative propositions, such as the fact that the sum of three and four is seven. In contrast, rules represent procedural knowledge of how to do things and are executed to produce actions (Squire, 2004). Rules are coded in ACT-R as condition-action (if-then) pairs. For example, a simple rule could be, if seeing a light is illuminated, then press a button. For chunks, ACT-R has algorithms for calculating chunk retrieval time and retrieval probability based on chunk activation and association in its declarative module, which stores and retrieves declarative knowledge. Larger values of chunk activation lead to faster and more successful retrieval. For rules, ACT-R has algorithms for rule selection and learning based on rule utility in its production module, which matches, selects, and executes rules (for details, see Anderson et al., 2004; Anderson and Lebiere, 1998). These algorithms are part of the general architecture that is the same for modelling different tasks, whereas task-specific knowledge of chunks and production rules need to be specified individually for each task in each model. Previous studies using ACT-R have modelled complex cognitive tasks including reading comprehension (Anderson et al., 2001), decision making (Fu and Anderson, 2006), and language learning (Taatgen and Anderson, 2002). Since QN and ACT-R possess unique and complementary modelling capabilities, the integration of the two architectures can benefit from their strengths and overcome the limitations of each one.

Concurrent goal scheduling

In ACT-R, each module has a buffer as an interface to connect with other modules. One of ACT-R's theoretical positions is that each buffer (e.g., the goal buffer storing the information about the current task) can hold only a single chunk at a time (Anderson et al., 2004). This position often requires dual-task models in ACT-R to combine the two goals from the two task components into one task-specific goal for the dual-task in question (e.g., Byrne and Anderson, 2001). This task-specific modelling method makes it difficult to model a wide range of dual-task scenarios, because modellers must define the task-specific knowledge for each scenario. From the QN perspective, multiple goals can co-exist in the goal buffer, and multi-task performance emerges as the behaviour of multiple streams of information flowing through a network, with no need for multitask-specific goals to interleave production rules into a serial programme or for an executive process to interactively control task processes (Liu, 1997; Liu et al., 2006).

Encouragingly, this QN position has been adopted in recent ACT-R based threaded cognition work for multi-task scheduling (Salvucci and Taatgen, 2008). In fact, the core mechanism of threaded cognition can be represented as a special type of queueing with a serial server (i.e., the production module in ACT-R) that gives priority to the longest waited task entity. The full integration of QN and ACT-R can further test and examine different queueing scheduling mechanisms to model a wider range of multi-task scenarios, especially those involving dynamic and complex cognitive tasks and producing high mental workload.

Module jamming

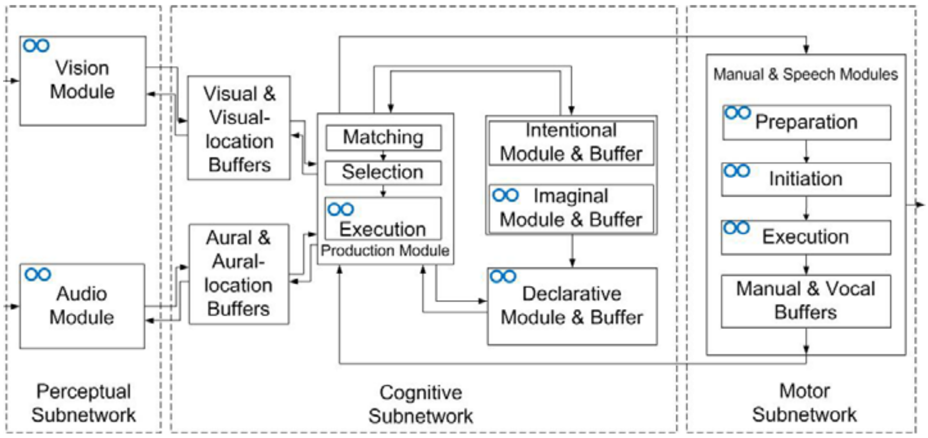
Another theoretical assumption in ACT-R is that a module processes one request at a time, and its buffer content is limited to a single declarative chunk (ACT-R Group, 2011; Anderson et al., 2004). As a result of this serial assumption, if a module is busy processing a request and receives another request at the same time, the module will be jammed. To avoid this jamming issue, ACT-R modelling requires modellers to "query the state in every production that makes a request that could potentially jam a module" [ACT-R Group, (2011), p.9]. From the QN perspective, this module jamming issue can be naturally resolved by adding queues to the modules. When a request arrives at a busy module, the request can wait in a queue until the module is free. This QN method does not require a production rule to query the state of the module or wait for the process in the module to finish, and therefore it is more suitable to model fast motor performance such as skilled transcription typing. The existence of queues in the motor processors has also received neurophysiological support from previous research (Roland et al., 1980).

The value of integrating QN and ACT-R in multitask modelling is demonstrated in this paper, which reports our completed work of integrating QN and ACT-R architectures. The resulting architecture is named QN-ACTR. We first introduce the theoretical assumptions and implementation of QN-ACTR. Then we describe the verification of QN-ACTR in comparison to the same tasks that have been modelled in ACT-R. Finally, we illustrate the benefits of the integration in the simulation of transcription typing tasks involving multi-task performance and reading comprehension, demonstrating its improved modelling capabilities in complex cognitive and multi-task scenarios. The integration allows QN-ACTR to model skilled typing performance and avoid jamming in the motor module, without the need for a production rule to frequently query the motor module state before each keystroke. The integration also enables QN-ACTR to model the complex cognition involved in reading comprehension that cannot be modelled by the QN architecture alone.

2 Queueing network-adaptive control of thought rational (QN-ACTR)

Figure 1 shows the current mental structure of QN-ACTR that results from merging QN and ACT-R architectures. The servers in QN-ACTR correspond to ACT-R's modules and buffers, some of which are grouped to match the corresponding servers of the QN structure as in QN-MHP. Entities travelling between these servers correspond to ACT-R's information units including buffer requests, chunks, and production rules. Table 1 shows the functional correspondence between ACT-R modules and QN servers that supports the integration of their structures. The processing logics in each QN-ACTR server are identical to the algorithms in the corresponding ACT-R module, including the sub-symbolic computations in the production and the declarative modules. As previously described, ACT-R modules do not have queueing mechanisms, but queues can be added from the QN perspective to support the scheduling of multi-tasks at the local server level. In QN-ACTR, queues are added to the modules that have non-zero processing time and a limited capacity. Currently, we do not apply any constraint to the capacity of queues and assume no time is needed for an information entity to enter or leave a queue. These assumptions are the same as the ones in previous QN modelling work (Liu et al., 2006; Wu and Liu, 2008a).

Figure 1 Server structure of QN-ACTR (see online version for colours)



Note: Queue symbols (shown as two circles) mark the servers where queues are added from the QN's perspective. All the server processing logics in the QN-ACTR are identical to the corresponding algorithms in ACT-R

Source: adapted from Cao and Liu (2012b)

Table 1 Correspondence between ACT-R modules and QN servers

ACT-R modules and buffers	*Corresponding QN servers. See (Wu and Liu, 2008a) for server details.*
Vision module	Server 1–4
Audio module	Server 5–8
Visual and visual-location buffers	Server A
Aural and aural-location buffers	Server B
Production module	Server C, D, F
Intentional and imaginal modules and buffers	Server E, G
Declarative module and buffer	Server H
Manual and speech modules and buffers	Server V, W, X, Y, Z, 21–25

Source: from Cao and Liu (2012b)

As shown in Figure 1, the servers are mainly based on ACT-R's modules and buffers. In ACT-R, there is no special module of working memory. Instead, "working memory can be equated with the portion of declarative memory above a threshold of activation"

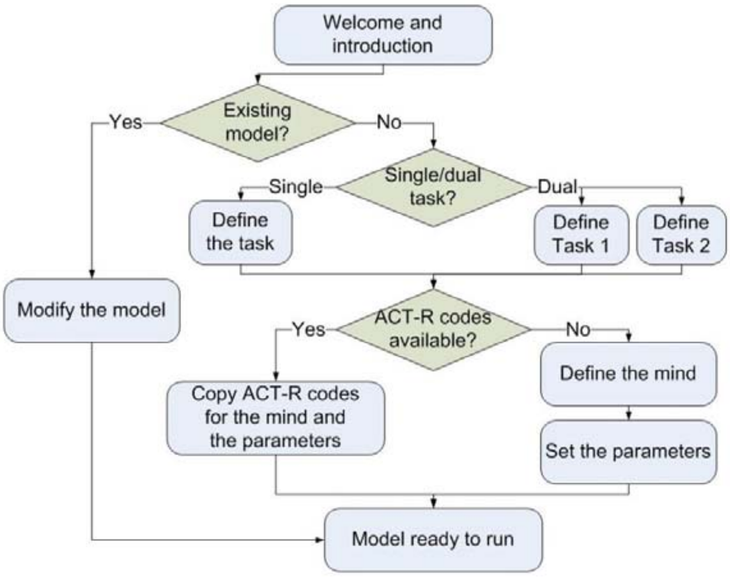
[Anderson et al., (1996), p.221], and the chunks that are temporarily held in the cognitive subnetwork (e.g., in the goal buffer) may provide a context and spread activation to the chunks in declarative memory, facilitating the retrieval of context-relevant declarative chunks. In this regard, entities (e.g., chunks) in the queues of the cognitive subnetwork may provide sources of activation for working memory. In the perceptual subnetwork, there may be a connection between the queues and short term sensory storages, including the visual sensory memory (Dick, 1974) and the auditory sensory memory (Darwin et al., 1972), because they both have the functionality of temporarily storing information. In the motor subnetwork, queues are believed to temporarily store time-ordered motor commands (Roland et al., 1980). The correspondence between queues and cognitive or neurological constructs, however, is not the focus of the current study. Instead, we focus on the verification and validation of QN-ACTR in terms of human performance modelling capabilities and demonstrate the importance of the theoretical concepts of 'queuing' in understanding cognitive architecture and multi-task performance.

At the implementation level, QN-ACTR is a computerised programme built on a discrete event simulation platform Micro Saint® Sharp (http//www.maad.com). A full integration was implemented, which means that all the data structures and the functions in ACT-R have been ported from the original Lisp implementation to Micro Saint® Sharp. This programming platform is selected because of three major reasons. First, this platform provides graphical interfaces for easy QN construction and simulation and also supports the visualisation of server network and task interaction. Second, this C# based platform supports C# programming plug-in functions and the connection with other C# applications, which help the development of QN-ACTR as an easy-to-use cognitive engineering tool with features such as model building assistants and human experiment platforms. Third, it is the same platform on which IMPRINT (Allender et al., 1997) is implemented and therefore supports future seamless integration between QN-ACTR and IMPRINT – a widely used system and task analysis tool that has its unique features in mental workload modelling.

For modellers using QN-ACTR, the model development process includes the steps of task setup, mind setup, and parameter setup (Figure 2). The task setup refers to the specification of display and control mechanisms in the task or experiment. The mind setup refers to the specification of task-specific knowledge, i.e., declarative chunks and production rules. The parameter setup adjusts parameters that control the model performance. QN-ACTR provides two methods to build a model – a text-based syntax method and a click-and-select interface (Cao and Liu, 2012a). The syntax method supports fast and direct model editing (i.e., copy and paste), which is designed for advanced users. Since the syntaxes for the mind and the parameter setups are identical to ACT-R syntaxes, available ACT-R codes can be directly used for QN-ACTR models. The click-and-select interface assists novice users and allows them to describe the model following experiment logic and mental knowledge by selecting from menu items and filling in blanks using a natural language (English), without the need to learn any special programming or cognitive engineering language.

QN-ACTR's modelling results include the trace of simulated mental activities, behavioural responses, reaction times, correct rates, and mental workload. These results can be visualised while the model is performing the task and stored in data sheets for future analyses. QN-ACTR also has an integrated human experiment platform for human data collection. For example, a modeller can use QN-ACTR to conduct simulated driving experiments with steering wheels and pedals. This feature allows models and humans to perform and be compared in the same tasks with identical interfaces, with no need to replicate the real world experiment system in the modelling platform for models to interact with. Using the same experiment platform saves programming labour and, more importantly, avoids any discrepancy between human and model tests caused by different experiment setups.

Figure 2 Flow chart showing the model development process in QN-ACTR (see online version for colours)



Source: from Cao and Liu (2012a)

3 Model verification

3.1 Method

After merging the structures and functions of QN and ACT-R architectures, we conducted a verification study with the purpose of examining whether the ACT-R functions built in QN-ACTR are accurate and complete. We tested QN-ACTR in the simulation of 20 tasks (summarised in Table 2) that have been modelled by ACT-R and threaded cognition. Among these tasks, 17 tasks were selected from ACT-R 6.0 (v1.3) tutorial to cover a wide range of typical single-tasks modelled in ACT-R, including visual-motor, auditory-vocal, declarative learning, procedural learning, and decision making tasks. The other three tasks were selected from dual-task experiments (Schumacher et al., 2001) using the psychological refractory period paradigm and have been modelled by threaded cognition implemented in ACT-R. As we have previously described, threaded cognition can be considered as a special type of queueing mechanism applied to ACT-R's production module, giving priority to the longest waited task. All together, these 20 tasks provide a thorough test bed to verify the implementation and programming of QN-ACTR.

Table 2 Descriptions of the 20 tasks modelled to verify QN-ACTR

Model	*Description*
	ACT-R 6.0 (v1.3) tutorial models
Addition	Compute 5 + 2 by counting 5, 6, and 7.
Count	Count from 2 to 4.
Semantic	Judge if an animal belongs to a category.
Tutor-model	Compute 36 + 47 by first adding the single digits and then the ten digits.
Demo	See a letter. Press the key for the letter.
Unit-2-assignment	See three letters. Two are the same. Press the key of the single letter.
Sperling	Briefly see 12 letters in three rows. Press keys for letters in the target row, which is indicated by the pitch of a tone. Higher pitch = higher row.
Subitize	See several 'x's. Say how many 'x's there are.
Paired	Memorise and recall 20 word-number pairs, one pair a trial.
Zbrodoff	Judge alphabetic arithmetic problems by pressing keys. For example, A + 2 = C is correct, but B + 3 = F is not.
Fan	Remember a set of people-location facts. Answer queries such as 'is the captain in the park?'
Group	Imperfect recall of 9 numbers in three groups, 123-456-789.
Siegle	Imperfect single-digit addition due to number similarity.
Bst-learn	Create a stick of a particular target length by selecting building sticks with three different lengths using the mouse.
Choice	Repetitively guess a biased coin.
Paired-learning	Same as the paired model except starting the task-specific knowledge from descriptive instructions instead of procedural rules.
Past-tense	Learn English past tenses from examples, demonstrating the overregularisation of irregular verbs.
	Schumacher et al. (2001) experiments modelled by threaded cognition
Exp. 1	Psychological refractory period, visual-motor and auditory-vocal dual-task. No specific response order.
Exp. 2	Same as Exp. 1 except prioritising on the auditory-vocal task.
Exp. 3	Same as Exp. 1 except using incompatible stimulus-response associations instead of compatible ones.

Source: adapted from Cao and Liu (2011a)

Models for the 20 tasks were developed strictly following their corresponding ACT-R models. The task displays and controls were modelled using QN-ACTR's syntaxes and templates. The codes specifying task-specific knowledge and parameters were directly copied from the codes used in the original ACT-R models. In addition, we also used the Common Random Numbers technique (McGeoch, 1992) to assign ACT-R and QN-ACTR models the same randomisation method and the same seeds in order to further control the sources of variance in the verification. Table 3 shows excerpts from the

QN-ACTR model codes for the Demo task. This task presents a random letter on the screen and requires a key press response corresponding to the displayed letter.

Table 3 Model code excerpts from a QN-ACTR model for the demo task in ACT-R's tutorial

Model setup step	*Model codes*
Task description	(use_task_dbt_template :method discrete_display_feedback_two_stages_method) (add_trials_from_discrete_display_feedback_two_stages_method :add_number_of_trials_per_block 1 :number_of_responses_per_trial 1 (:item_type display_item_visual_text :visual_text ("B" "C" "D" "F" "G" "H" "J" "K" "L" "M" "N" "P" "Q" "R" "S" "T" "V" "W" "X" "Y" "Z") :correct_response_to_each_visual_text (b c d f g h j k l m n p q r s t v w x y z) :text_randomisation without_replacement :display_item_screen_location_x (125) :display_item_screen_location_y (150)))
Declarative knowledge description	(chunk-type read-letters state) (chunk-type array letter) (add-dm (start isa chunk) (attend isa chunk) (respond isa chunk) (done isa chunk) (goal isa read-letters state start))
Procedural knowledge description	(P find-unattended-letter =goal> ISA read-letters state start ==> +visual-location> ISA visual-location :attended nil =goal> state find-location)
Parameters	(sgp :v t :needs-mouse nil :show-focus t :trace-detail high)

Note: The syntaxes specifying task-specific knowledge and parameters are identical to the ones used in ACT-R

3.2 Results

Both the mental activity results recorded in the model output traces and the behavioural results such as reaction times and correction rates were compared between QN-ACTR and ACT-R models. Output traces were compared line by line to examine both the time and event contents. For example, the following line of trace, 222.423 DECLARATIVE RETRIEVED-CHUNK pair18-0

from the paired model results in QN-ACTR showed that at clock time 222.423 second, the model's declarative module retrieved the chunk *pair18-0*. This trace was identical to the corresponding ACT-R trace. For the tasks with quantitative behavioural results, mean absolute percentage error (*MAPE*), root mean square error (*RMSE*), and coefficient of determination (R^2) were computed between QN-ACTR and ACT-R results.

Table 4 QN-ACTR verification results

Model	*Results*
	ACT-R 6.0 (v1.3) tutorial models
Addition	Same output traces ended at 0.5 s.
Count	Same output traces ended at 0.3 s.
Semantic	Same output traces for test 1 (ended at 0.15 s), test 2 (0.25 s), and test 3 (0.35 s).
Tutor-model	Same output traces ended at 0.45 s.
Demo2	Same output traces for both home-location key condition (ended at 0.785 s) and other key condition (1.035 s).
Unit-2-assignment	Same output traces for both home-location key condition (ended at 1.055 s) and other key condition (1.305 s).
Sperling	1,000 run results (numbers of correct responses): *MAPE* = 1.4%, *RMSE* = 0.032, R^2 = 0.997.
Subitize	Same output traces. 1 run results (both reaction time and correction rate): *MAPE* = 0%, *RMSE* = 0, R^2 = 1.
Paired	
Zbrodoff	
Fan	
Group	Same output traces ended at 18.535 s.
Siegle	Same output traces. 500 run results (response distribution): *MAPE* = 0%, *RMSE* = 0, R^2 = 1.
Bst-learn	Same output traces. 5 run results (overshoot chances and rule utilities): *MAPE* = 0%, *RMSE* = 0, R^2 = 1.
Choice	Same output traces. 1 run results (rate of guessing heads): *MAPE* = 0%, *RMSE* = 0, R^2 = 1.
Paired-learning	Same output traces. 1 run results (both reaction time and correction rate): *MAPE* = 0%, *RMSE* = 0, R^2 = 1.
Past-tense	Same production rules composed and same output traces over ten trials, except for a 0.001% time difference.
	Schumacher et al. (2001) experiments modelled by threaded cognition
Exp. 1	1 run results (reaction time): *MAPE* = 1.3%, *RMSE* = 7.554, R^2 = 0.998.
Exp. 2	30 run results (reaction time): *MAPE* = 2.3%, *RMSE* = 10.851, R^2 = 0.983.
Exp. 3	3 run results (reaction time): *MAPE* = 4.0%, *RMSE* = 37.742, R^2 = 0.939.

Source: adapted from Cao and Liu (2011a)

As summarised in Table 4, the verification results showed that QN-ACTR models generated the same results as the original ACT-R models. Results from 15 of the 20 tasks were identical between QN-ACTR and ACT-R. For the other five models, results were very similar (*MAPE* < 5.0% and R^2 > 0.9). The sources of the remaining variances include the difference of built-in random number functions between Lisp and C#, which were used in randomly focusing visual attention on the next item, and the difference in rounding digits between Lisp and C#. These results concluded that the ACT-R functions built in QN-ACTR are accurate and complete.

4 QN-ACTR simulation of transcription typing and reading comprehension tasks

The verification of QN-ACTR described in the previous section established the basis for further incorporating unique QN mechanisms into QN-ACTR. In this section, we demonstrate the improved modelling capability of QN-ACTR after adding QN mechanisms. A model was built using QN-ACTR to simulate transcription typing tasks involving dual-task performance and reading comprehension, illustrating the benefits of the integrated cognitive architecture in modelling complex cognition and multi-task scenarios and resolving the concurrent goal scheduling and the module jamming issues in ACT-R. The transcription typing tasks were selected, because

1 previous studies about transcription typing have accumulated numerous detailed empirical results that are very useful for comparing models

2 the previous QN architecture (i.e., QN-MHP) has modelled many transcription typing phenomena but has difficulty in modelling the phenomena that involve reading comprehension

3 skilled transcription typing and related dual-task scenarios may cause the concurrent goal scheduling issue and the module jamming issue in ACT-R and have not been modelled in ACT-R.

4.1 Method

Transcription typing is one of the most common activities in human-computer interaction. It involves complex interaction of perception, cognition, and motor processes in the scale of tens to hundreds milliseconds. A considerable amount of phenomena in transcription typing has been studied and documented, including basic behavioural performance and the effects of skills, typing contents, and concurrent tasks (Gentner, 1983; Inhoff and Wang, 1992; Rayner, 1998; Salthouse, 1984, 1986; Salthouse and Saults, 1987). It has been regarded as one of the major tasks to test cognitive architectures (Newell, 1990). Several qualitative and quantitative models have been developed to model transcription typing, and a recent study using QN-MHP has modelled most of the phenomena in the literature but cannot model the two phenomena that involve reading comprehension (Wu and Liu, 2008b).

The integrated QN-ACTR cognitive architecture can model reading comprehension using ACT-R's declarative memory mechanisms. In ACT-R, sentence memory can be modelled as declarative chunks of syntactic and semantic representations, and reading comprehension can be modelled as memory retrieval and inference/interpretation (Anderson et al., 2001; Budiu and Anderson, 2004; Lewis and Vasishth, 2005). Also, QN-ACTR can model transcription typing and concurrent task scheduling using QN mechanisms that have been demonstrated in QN-MHP. These mechanisms are not included in ACT-R but are necessary for modelling skilled transcription typing phenomena and dual-task coordination.

First, queueing mechanisms coordinate multi-task performance at the local server level without the need to define any multitask-specific knowledge or any executive process. In QN-ACTR, the goal buffer of the intentional module can hold multiple goal chunks simultaneously representing the concurrence of multiple task components, and a sorted queue is implemented in the production module to coordinate concurrent tasks. The queue consists of entities each representing a task component and sorts entities based on their waiting time. The waiting time of a task component is initialised as zero at the beginning of the simulation and is reset to zero each time when the task component is processed by the production module executing a production rule. If multiple task components are competing for the limited production module resource, the task component closest to the front of the queue will receive priority. In the motor subnetwork, first-in-first-out queues are implemented in the servers of preparation, initiation, and execution. These servers explicitly represent the three processing stages in ACT-R's motor module, but in ACT-R they have no queue. Without any queue, an ACT-R model would execute a production rule and query the state of the manual module before typing each key to avoid jamming the module (e.g., the Sperling model in ACT-R Group, 2011). This modelling method has difficulty in modelling fast and skilled typing performance. In QN-ACTR, a production rule can execute a manual action to type in the unit of a word. Motor servers still process one letter at a time, but extra letters in a word can wait in a queue, as a way to model skilled typing performance and avoid the module jamming issue.

Second, parallel processing is created in the motor execution server, allowing parallelism between individual motor effectors. For example, a hand can move simultaneously with another hand or a foot. This parallel processing has not been included in ACT-R but is necessary for modelling the parallelism of motor movements evidenced in transcription typing. For instance, a typist's concurrent task of pressing a foot pedal as soon as they heard a tone did not affect typing performance (Salthouse and Saults, 1987). In addition, successive keystrokes from fingers on alternate hands are faster than successive keystrokes from fingers on the same hand (Wu and Liu, 2008b). The motor servers of QN-ACTR follow the corresponding parallel processing implementation in QN-MHP.

Third, the learning effect on server processing time is mathematically modelled in two motor servers – preparation and initiation, modelling motor learning. For example, one of the typing phenomena shows that repetitive one-finger tapping time decreases with typing skills (Salthouse, 1984). In the QN architecture, the effect of motor skill learning on reaction time is modelled mathematically using an exponential function (Feyen, 2002). Although the power function was also used to model practice and learning effects (Anderson et al., 2009; Newell and Rosenbloom, 1981), research has shown that the exponential function is a better candidate for the law of practice than the power function (Heathcote et al., 2000). As a result, in QN-ACTR, the server processing time (T_i) is modelled as an exponential function of the number of entities (N_i) processed by the server (i),

$$T_i = A_i + B_i \, Exp(-\alpha_i N_i) . \quad (1)$$

A_i represents the expected minimal processing time after intensive practice; B_i is the change in the expected processing time from the beginning to the end of practice; α_i represents the learning rate. This equation has been used in QN-MHP and successfully modelled the effects of motor skill learning in psychological refractory period experiments (Wu and Liu, 2008a) and transcription typing experiments (Wu and Liu, 2008b). QN-ACTR has implemented this equation in two motor servers to model motor learning. Currently, the four parameters in this equation are the only parameters that have been integrated into QN-ACTR from the QN architecture. The other parameters in QN-ACTR are from the ACT-R architecture. Detailed descriptions are not included here due to the limited space but can be found in the ACT-R reference manual (ACT-R Group, 2011).

The integrated QN-ACTR architecture was tested in the simulation of 29 transcription typing phenomena, particularly the two phenomena involving reading comprehension and a phenomenon involving concurrent tasks. The descriptions and empirical results of these phenomena are summarised in Table 5. Salthouse (1984) tested skilled typists using the Nelson-Denny Reading Test and found that the typing speed in the typing-and-reading condition (58 words-per-minute) was much slower than the reading speed in the reading-only condition (253 words-per-minute). The accuracy of reading comprehension was lower in the typing-and-reading condition (44.7%) compared with the reading-only condition (58.1%). The typing interkey time (177 ms) in the typing-and-reading condition was similar to the time in the typing-only condition (181 ms; Salthouse and Saults, 1987). In another study, the author found that the correlation between typing speed and comprehension scores obtained when typing was not significant (Salthouse, 1986) and concluded that the typing skill and the comprehension skill are independent. Another important phenomenon is that a concurrent task does not affect typing performance. Salthouse and Saults (1987) added a secondary task in parallel with the primary task of transcription typing. Instructions asked typists to press a foot pedal as soon as they heard a tone signal but prioritise the typing task as the primary task. The results showed that this concurrent auditory-pedal task did not affect typing performance, as the typing interkey time was 185 ms in the concurrent-task condition versus 181 ms in the typing-only condition.

In the model, the task displays and controls were coded using QN-ACTR's task template for discrete and trial-based experiments. The declarative and procedural knowledge for typing was modelled following the Demo model in ACT-R Tutorial Unit 2. Four production rules were defined to model the procedure of typing, including *find-unattended-word*, *attend-word*, *encode-word*, and *type-word*. As we previously described, queues in the motor servers allow skilled typing performance to be modelled in the unit of a word rather than a letter. Reading comprehension was modelled following similar models in ACT-R (Anderson et al., 2001). When reading a word in a sentence, the model retrieves the semantic meaning of the word. After the meaning of the word is retrieved, it is stored in a slot of a chunk representing the semantic meaning of the sentence. After finishing a sentence, the model memorises the semantic chunk of the sentence in its declarative memory. Several comprehension questions in the form of propositions are asked after the model finishes reading a passage. For each proposition, the model searches its declarative memory for any chunk encoding the related semantic information. The model can answer the comprehension question correctly if the retrieval succeeds, but it cannot answer the question if the retrieval fails. Since memory limitations such as forgetting are modelled in the sub-symbolic computations of the declarative module, the model can capture comprehension errors caused by forgetting. For the concurrent task phenomena, the secondary auditory-pedal task was modelled following the Sperling task in ACT-R's tutorial. When a tone is presented, the model first detects the sound and then responds by issuing a pedal-pressing action. As we previously described, the goal chunk of this secondary task co-exists with the goal chunk of the primary typing task in the goal buffer. The queueing mechanism in the production module coordinates multiple tasks without the need to define any multitask-specific knowledge or any executive process. Since the primary typing task was stressed by instructions in the original experiment, the typing task was assigned a higher priority than the auditory-pedal task in the queueing mechanism. All related parameters used values from previous studies (Table 6), with all other parameters at their default values. As previously described in equation (1), the parameters A_i, B_i, α_i, and N_i affect how the motor servers' processing time decreases with practice. The other parameters are integrated from the ACT-R architecture, and Table 6 provides a brief description (see ACT-R reference manual for details, ACT-R Group, 2011). The Nelson-Denny Reading Test was used in the simulation as in the empirical experiments.

Table 5 Summary of transcription typing phenomena modelled in QN-ACTR

Phenomena description	*Empirical human results*	*QN-ACTR simulation results*
	Phenomena involving complex cognition	
Typing is slower than reading.	Reading speed = 253 words-per-minute (wpm); typing speed = 58 wpm; comprehension accuracy is 58.1% for reading only and 44.7% for reading while typing. Results from 74 typists typing or reading passages with about 1,200 characters. (Salthouse, 1984)	Reading speed = 267 wpm (absolute percentage error, *APE* = 5.4%); typing speed = 69 wpm (*APE* = 19.0%); comprehension accuracy is 55.3% (*APE* = 4.9%) for reading only and 44.9% (*APE* = 0.4%) for typing and reading.
Typing skill and comprehension are independent.	No significant correlation (i.e., $r = -0.169$, $p > 0.15$). 74 typists. Passage with about 1200 characters. (Salthouse, 1986)	The correlation $r = 0.042$, $p = 0.80 > 0.15$.
	Dual-task phenomena	
A concurrent task does not affect typing performance.	The interkey time in the concurrent task situation (185 ms) was not significantly longer than that in single-task typing (181 ms). 40 typists. Passage with about 1,250 characters. (Salthouse and Saults, 1987)	Interkey time in the concurrent task situation was 177 ms (*APE* = 4.3%), similar to typing only (172 ms).
	Other phenomena	
Typing is faster than choice reaction time.	Typing interkey time (median) = 177 ms for skilled typists; choice reaction time = 560 ms. 74 typists. Passage with about 1200 characters. (Salthouse, 1984)	Typing interkey time (median) = 182 ms (*APE* = 2.8%); choice reaction time = 495 ms (*APE* = 11.7%).

Table 5 Summary of transcription typing phenomena modelled in QN-ACTR (continued)

Phenomena description	*Empirical human results*	*QN-ACTR simulation results*
	Other phenomena	
Typing rate is independent of word order.	Qualitative phenomena (Wu and Liu, 2008b)	Typing interkey time = 172 ms for normal order and 172 ms for randomised word order, t (6601) = 0.001, p = 0.999.
Typing speed is slower with random character order.	Interkey time in typing increased to 454 ms when typing materials composed of words with random characters. Five subjects (three typists). 220 words. (Hershman and Hillix, 1965)	Typing interkey time = 172 ms for normal order and 373 ms for random character order (*APE* = 17.8%).
Typing rate is impaired by restricted preview.	Typing rate decreases with smaller preview window of the material to be typed. Eight typists. Six passages each with about 74 words. (Inhoff and Wang, 1992)	R^2 of simulated interkey time is 0.98 (*APE* = 9.2%).
Alternate-hand advantage	Alternate-hand keystrokes are about 45 ms faster than the same-hand keystrokes (Wu and Liu, 2008b)	78 ms faster (*APE* = 73.6%).
Digram frequency effect	Digram (letter pairs) that occur more frequently in normal language are typed faster than less frequent digram. 45 typists. Passage with about 1,250 characters. (Salthouse, 1984)	Significantly faster, t (197) = –11.062, p < 0.001.
Interkey time is independent of word length.	No significant difference between long and short words. 74 typists. Passage with about 1,200 characters. (Salthouse, 1984)	No significance, t (387) = 0.381, p = 0.70.
Word initiation effect	The first keystroke in a word is about 45 ms slower than the subsequent keystrokes. 74 typists. Passage with about 1,200 characters. (Salthouse, 1984)	53 ms slower (*APE* = 17.1%).
Context phenomenon	The time for a keystroke is dependent on the specific context in which the character appears, especially keyboard topography (Wu and Liu, 2008b)	Interkey time for the same key can range from around 60 ms to 200 ms, depending on the context of the previous key.
Copying span	14.6 characters. 29 typists. Eight sentences, each about 75 characters. (Salthouse, 1985)	10.9 characters (*APE* = 25.5%).
Stopping span	2.16 characters. 12 secretaries. 300 sentences, each about 28 characters. (Logan, 1982)	1.8 characters (*APE* = 16.7%).

Table 5 Summary of transcription typing phenomena modelled in QN-ACTR (continued)

Phenomena description	*Empirical human results*	*QN-ACTR simulation results*
	Other phenomena	
Eye-hand span	5.25 characters, averaged from multiple studies (for details, see Salthouse, 1986).	6.1 characters (*APE* = 16.0%).
Eye-hand span is smaller for randomly ordered letters.	1.75 characters. 74 typists. Passage with about 1,200 characters. (Salthouse, 1984)	1.0 characters (*APE* = 42.9%).
Replacement span	2.8 characters. 85 typists. Passages with about 1,200 characters. (Salthouse and Saults, 1987)	4.5 characters (*APE* = 60.7%).
Detection span	8.1 characters. 85 typists. Passages with about 1,200 characters. (Salthouse and Saults, 1987)	9.1 characters (*APE* = 12.3%).
Two-hand digrams or two-finger digrams exhibit greater changes with skills than do one-finger digrams.	The slope of the regression equations relating the digram interval (ms) to typing speed of two-hand digrams (–2.08) and two-finger digrams (–2.38) were greater than that of one-finger digrams (–1.38). 74 typists. Passage with about 1,200 characters. (Salthouse, 1984)	Two-hand (–6.05) and two-finger (–4.06) are greater than one finger (–2.92 on average).
Repetitive tapping rate increases with skill.	Significant positive correlation between the tapping rate and the net typing speed ($p < 0.01$). 74 typists. Passage with about 1,200 characters. (Salthouse, 1984)	$r = 0.81$ ($p < 0.01$).
The variability of interkey time decreases with the skills.	Inter-keystroke variability correlated –0.69 with the net typing speed; intra-keystroke variability correlated –0.71 with the net typing speed. 74 typists. Passage with about 1,200 characters. (Salthouse, 1984)	Inter-key $r = -0.85$ ($p < 0.05$); intra-key $r = –0.90$ ($p < 0.05$)
Eye-hand span is larger with increased skills.	The correlation between the eye-hand span and net words-per-minute was significant with $p < 0.01$. 74 typists. Passage with about 1,200 characters. (Salthouse, 1984)	$r = 0.99$ ($p < 0.05$).
Replacement span is larger with more skills.	The correlation between net words per minute and the replacement span was 0.80 ($p < 0.01$). 29 typists. Eight sentences, each about 75 characters. (Salthouse, 1985)	$r = 0.61$ ($p < 0.05$).
Interkey time decreases with practice.	Qualitative phenomena (see Gentner, 1983)	$R^2 = 0.94$ with significant correlation, $p < 0.05$.
Eye gaze duration-per-character decreases with increased preview window size.	(see Figure 2 in Wu and Liu, 2008b).	R^2 of the simulated fixation time is 0.97 (*APE* = 21.6%).

Table 5 Summary of transcription typing phenomena modelled in QN-ACTR (continued)

Phenomena description	*Empirical human results*	*QN-ACTR simulation results*
	Other phenomena	
Eye saccade size	Four characters, averaged from multiple studies (for details, see Rayner, 1998).	4.1 characters (*APE* = 2.5%).
Eye fixation duration	400 ms, averaged from multiple studies (for details, see Rayner, 1998).	705 ms (*APE* = 76.3%).

Table 6 Descriptions, values, and sources of parameters used in the transcription typing model

Parameter	*Description*	*Value and source*
α_i	Learning rate alpha	0.001 (Heathcote et al., 2000; Wu and Liu, 2008b)
A_i	Expected minimal processing time after intensive practice	21.5 ms (Anderson and Lebiere, 1998; Card et al., 1983)
B_i	Change of expected processing time from the beginning to the end of practice	50 ms (Anderson and Lebiere, 1998; Card et al., 1983)
N_i	Total number of digrams that have been processed	15,000,000 for skilled typists (Wu and Liu, 2008b)
:imaginal-delay	Determine the time for the imaginal module to form a chunk of imaginal representation.	0.100 s (Mehlhorn and Marewski, 2011)
:lf	Latency factor for declarative retrieval time. Larger values lead to longer memory retrieval time.	0.003 (Budiu and Anderson, 2004)
:bll	Base level learning parameter for chunk activation. Larger values lead to faster activation decay.	0.3 (Pavlik and Anderson, 2005)
:rt	Retrieval threshold. Set the minimum activation a chunk must have to be able to be retrieved.	–0.704 (Pavlik and Anderson, 2005)
:ans	Set the instantaneous noise added to chunk activation.	0.5 (Anderson and Matessa, 1997)
:tone-recode-delay	Determine the auditory perception time to recode a tone sound.	0.05 s (Byrne and Anderson, 2001)
saccade duration	Saccade movement duration	20 ms for saccade execution, plus an additional 2 ms for each degree of visual angle (Salvucci, 2001)

Note: The first four parameters are from the QN-MHP architecture, whereas the other parameters are from the ACT-R architecture

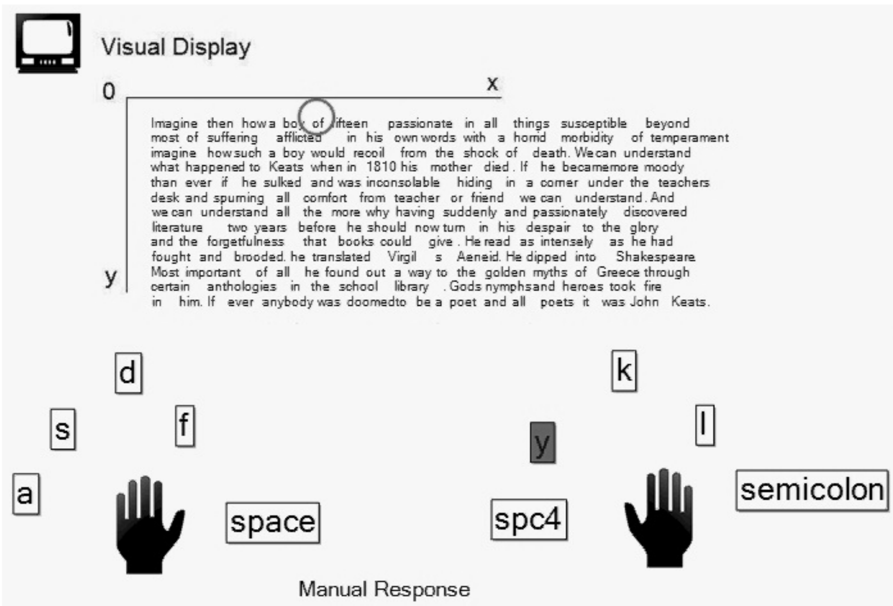
Source: adapted from Cao and Liu (2012b)

4.2 Results

The model simulated various transcription typing tasks and produced behavioural results. While the model is performing the task, the task visualisation feature in

QN-ACTR can show the simulated typing behaviour in real time, as illustrated in the screenshot of Figure 3. Modelling results include text output traces for module activities, typing performance such as finger movement and eye movement, and reading comprehension performance such as reading speed and comprehension accuracy. The absolute percentage error (*APE*) and the coefficient of determination (R^2) were computed between QN-ACTR's results and the human results. These results are summarised in Table 5. The modelling results were similar to the human results from the experiments. It is particularly important to note that the modelling results captured the phenomena involving reading comprehension (i.e., typing is slower than reading; typing skill and comprehension are independent) that is difficult to model by the QN architecture alone and the phenomena involving concurrent tasks and skilled typing (e.g., a concurrent task does not affect typing performance) that is difficult to model by ACT-R alone. To examine whether an ACT-R model without queues can produce similar results, we repeated the QN-ACTR simulation without the QN mechanisms that were introduced previously in Section 4.1 Method. In this case, the reduced version of QN-ACTR became the same as ACT-R, as demonstrated previously in Section 3 Model Verification. This simulation produced a much longer typing interkey time of 500 ms. In comparison, the result from the simulation with the QN mechanisms was just 182 ms, much closer to the human result of 177 ms. Removing the QN mechanisms did not change choice reaction time (still 495 ms).

Figure 3 The visualisation of the task interaction in QN-ACTR



Notes: The visual display section shows the texts on the screen and the location of visual attention (represented by a circle). The manual response section shows that the index finger of the right hand is pressing key 'y' while other fingers are resting at the home locations.

5 Discussion

QN-ACTR is an integrated cognitive architecture that unifies the QN mathematical architecture and the ACT-R symbolic architecture. Continuing the line of model integration, QN-ACTR combines QN's unique queueing mechanisms and hybrid server network and ACT-R's unique symbolic knowledge representations and sub-symbolic computations. This integration allows QN-ACTR to model a wider range of tasks, especially complex cognitive and multi-task scenarios.

The current study is our first step of QN-ACTR work. Before this study, it was unclear whether such integration is feasible. We focus on the verification of the integration and the demonstration of the improved modelling capability in the simulation of transcription typing and reading comprehension scenarios. The results show that the integrated QN-ACTR is able to model what have been modelled by ACT-R alone (e.g., cognitive tasks as summarised in Table 2) and QN-MHP alone (e.g., skilled typing). In addition, it can also model concurrent performance involving both typing and reading comprehension, which previous methods have difficulties to model. Reading comprehension was modelled following previous ACT-R models of sentence comprehension, which utilised ACT-R's advantages of the declarative memory and subsymbolic computations. Skilled typing was modelled following previous QN-MHP models, which utilised QN's advantages of queues and hybrid server network.

We also find that without the QN mechanisms, it is difficult for ACT-R mechanisms alone to simulate skilled typing performance. Without any queue in the motor subnetwork, the production module can only send motor typing commands in the unit of a single letter and wait for the completion of the previous typing action before issuing the next motor command. As a result, the typing interkey time became much longer (500 ms) than the human result (177 ms) and was similar to choice reaction time (495 ms for models; 560 ms for human), where responses were made one at a time. This comparison demonstrated the added value of QN-ACTR to human performance modelling and simulation.

The verification and validation results in the current study provide support for future work to examine other novel aspects of the integration, such as the use of queues in the perceptual modules. One of these aspects is to investigate the *buffer stuffing* issue. ACT-R implements a *buffer stuffing* mechanism that occurs for perceptual (visual and aural) modules and "is intended as a simple approximation of a bottom-up mechanism of attention" [ACT-R Group, (2011), p.11]. *Buffer stuffing* allows perceptual modules to process a stimulus without any 'top-down' request from a production rule, but it 'only occurs if the buffer is empty' [ACT-R Group, (2011), p.11]. Since ACT-R buffers have the capacity of one chunk, this mechanism has a limitation in modelling the 'bottom-up' processing in dynamic visual tracking tasks. When ACT-R tracks a visual object, such as a moving car, it keeps the representation of the object (i.e., a chunk) in the visual-location buffer until a production rule is fired to clear the buffer. The result, as discussed by Destefano (2010), is that the buffer is never empty and *buffer stuffing* for another visual object can never occur in such scenarios no matter how salient the other visual object is, for example, a flashing traffic light. Visual tracking modelled in ACT-R becomes "a closed-loop system that cannot be interrupted by any bottom-up activity" [Destefano, (2010), p.63]. From QN's perspective, such bottom-up perceptual activities may be temporarily stored in queues before further perceptual processing. In the current study, the queues in the visual and auditory modules were not involved in the typing and comprehension task, because the perceptual information (i.e., paragraphs for typing) was static and did not produce task demand beyond the processing capability of the visual module. Future studies can examine dynamic scenarios and explore the use of QN mechanisms to model interruptible visual tracking performance.

In addition, future studies can further investigate the cognitive and neurological bases of queues. There may be a connection between short term sensory storages and the queues of the perceptual modules, in the sense that the queues can temporarily store sensory information when the stimuli just disappear and the modules are still busy. The models in the current study did not use the queues in the visual and auditory modules, but the queues in the motor subnetwork were used in the simulation of typing and reading comprehension to store time-ordered motor commands of finger movement. As in previous QN modelling work, QN-ACTR currently does not apply any constraint to the capacity of queues. The models under this assumption have produced results similar to the human results. Future studies are needed to investigate the capacity of queues and its implication to human performance modelling.

In conclusion, QN-ACTR is an integrated cognitive architecture and a computerised simulation programme that combines the benefits of QN and ACT-R. ACT-R's modules and buffers are implemented as QN servers with their processing logics identical to the corresponding ACT-R algorithms. This QN representation of ACT-R has been verified in the simulation of 20 typical tasks from the ACT-R literature using the same task setups and codes from the original ACT-R models. From the QN perspective, three unique QN positions have also been implemented in QN-ACTR, including the queueing mechanisms to coordinate multi-task performance at the local server level, the hybrid server network to model parallel processing between individual motor effectors, and the mathematical function to model motor skill learning. The benefits of the integration have been demonstrated in the simulation of 29 transcription typing phenomena. In particular, QN-ACTR accounted for both the phenomena involving the complex cognitive activities of reading comprehension and the phenomena involving concurrent tasks and skilled typing, showing the improved modelling capability in complex cognitive and multi-task scenarios that have not been modelled by either QN or ACT-R.

References

ACT-R Group (2011) ACT-R 6.0 Tutorial, [online] http://act-r.psy.cmu.edu/.

Allender, L., Kelley, T., Archer, S. and Adkins, R. (1997) 'IMPRINT: the transition and further development of a soldier-system analysis tool', *MANPRINT Quarterly*, Vol. 5, No. 1, pp.1–7.

Anderson, A.M., Mirka, G.A., Joines, S.M.B. and Kaber, D.B. (2009) 'Analysis of alternative keyboards using learning curves', *Human Factors*, Vol. 51, No. 1, pp.35–45.

Anderson, J.R. and Lebiere, C. (1998) *The Atomic Components of Thought*, Erlbaum, Mahwah, NJ.

Anderson, J.R. and Matessa, M. (1997) 'A production system theory of serial memory', *Psychological Review*, Vol. 104, No. 4, pp.728–748.

Anderson, J.R., Bothell, D., Byrne, M.D., Douglass, S., Lebiere, C. and Qin, Y. (2004) 'An integrated theory of the mind', *Psychological Review*, Vol. 111, No. 4, pp.1036–1060.

Anderson, J.R., Budiu, R. and Reder, L.M. (2001) 'Theory of sentence memory as part of a general theory of memory', *Journal of Memory and Language*, Vol. 45, No. 3, pp.337–367.

Anderson, J.R., Reder, L.M. and Lebiere, C. (1996) 'Working memory: activation limitations on retrieval', *Cognitive Psychology*, Vol. 30, No. 3, pp.221–256.

Budiu, R. and Anderson, J.R. (2004) 'Interpretation-based processing: a unified theory of semantic sentence comprehension', *Cognitive Science*, Vol. 28, No. 1, pp.1–44.

Byrne, M.D. and Anderson, J.R. (1998) 'Perception and action', in J.R. Anderson and C. Lebiere (Eds.): *The Atomic Components of Thought*, pp.167–200, Lawrence Erlbaum, Mahwah, NJ.

Byrne, M.D. and Anderson, J.R. (2001) 'Serial modules in parallel: the psychological refractory period and perfect time-sharing', *Psychological Review*, Vol. 108, No. 4, pp.847–869.

Cao, S. and Liu, Y. (2011a) 'Integrating queueing network and ACT-R cognitive architectures', *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 55, No. 1, pp.836–840.

Cao, S. and Liu, Y. (2011b) 'Mental workload modeling in an integrated cognitive architecture', *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 55, No. 1, pp.2083–2087.

Cao, S. and Liu, Y. (2012a) 'An integrated cognitive architecture for cognitive engineering applications', *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 56, No. 1, pp.323–327.

Cao, S. and Liu, Y. (2012b) 'QN-ACTR modeling of multitask performance of dynamic and complex cognitive tasks', *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, Vol. 56, No. 1, pp.1015–1019.

Card, S., Moran, T.P. and Newell, A. (1983) *The Psychology of Human-Computer Interaction*, Lawrence Erlbaum Associates, Hillsdale, NJ.

Darwin, C.J., Turvey, M.T. and Crowder, R.G. (1972) 'An auditory analogue of the sperling partial report procedure: evidence for brief auditory storage', *Cognitive Psychology*, Vol. 3, No. 2, pp.255–267.

Destefano, M. (2010) *The Mechanics of Multitasking: The Choreography of Perception, Action, And Cognition Over 7.05 Orders of Magnitude*, Doctoral Dissertation, Rensselaer Polytechnic Institute.

Dick, A.O. (1974) 'Iconic memory and its relation to perceptual processing and other memory mechanisms', *Perception and Psychophysics*, Vol. 16, No. 3, pp.575–596.

Elkind, J.I., Card, S.K., Hochberg, J. and Huey, B.M. (1989) *Human Performance Models for Computer-Aided Engineering*, National Academy Press, Washington, DC.

Feyen, R. (2002) *Modeling Human Performance Using the Queuing Network-Model Human Processor (Qn-Mhp)*, Doctoral Dissertation, Department of Industrial and Operations Engineering, University of Michigan.

Fu, W-T. and Anderson, J.R. (2006) 'From recurrent choice to skill learning: a reinforcement-learning model, *Journal of Experimental Psychology: General*, Vol. 135, No. 2, pp.184–206.

Gentner, D.R. (1983) 'The acquisition of typewriting skill', *Acta Psychologica*, Vol. 54, Nos. 1–3, pp.233–248.

Heathcote, A., Brown, S. and Mewhort, D.J.K. (2000) 'The power law repealed: the case for an exponential law of practice', *Psychonomic Bulletin and Review*, Vol. 7, No. 2, pp.185–207.

Hershman, R.L. and Hillix, W.A. (1965) 'Data processing in typing: Typing rate as a function of kind of material and amount exposed', *Human Factors*, Vol. 7, No. 5, pp.483–492.

Holt, M. and Jahn, R. (2004) 'Synaptic vesicles in the fast lane', *Science*, Vol. 303, No. 5666, pp.1986–1987.

Inhoff, A.W. and Wang, J. (1992) 'Encoding of text, manual movement planning, and eye-hand coordination during copytyping', *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 18, No. 2, pp.437–448.

John, B.E. and Kieras, D.E. (1996) 'The GOMS family of user interface analysis techniques: comparison and contrast', *ACM Transactions on Computer-Human Interaction*, Vol. 3, No. 4, pp.320–351.

Laird, J.E. (2008) 'Extending the soar cognitive architecture', *Proceedings of the First Conference on Artificial General Intelligence (AGI-08)*.

Laird, J.E., Newell, A. and Rosenbloom, P.S. (1987) 'Soar: an architecture for general intelligence', *Artificial Intelligence*, Vol. 33, No. 1, pp.1–64.

Lewis, R.L. and Vasishth, S. (2005) 'An activation-based model of sentence processing as skilled memory retrieval', *Cognitive Science*, Vol. 29, No. 3, pp.375–419.

Liu, Y. (1996) 'Queueing network modeling of elementary mental processes', *Psychological Review*, Vol. 103, No. 1, pp.116–136.

Liu, Y. (1997) 'Queueing network modeling of human performance of concurrent spatial and verbal tasks', *IEEE Transactions on Systems, Man, Cybernetics*, Vol. 27, No. 2, pp.195–207.

Liu, Y. (2009) 'QN-ACES: integrating queueing network and ACT-R, CAPS, EPIC, and Soar architectures for multitask cognitive modeling', *International Journal of Human-Computer Interaction*, Vol. 25, No. 6, pp.554–581.

Liu, Y., Feyen, R. and Tsimhoni, O. (2006) 'Queueing network-model human processor (QN-MHP): a computational architecture for multitask performance in human-machine systems', *ACM Transactions on Computer-Human Interaction*, Vol. 13, No. 1, pp.37–70.

Logan, G.D. (1982) 'On the ability to inhibit complex movements: a stop-signal study of typewriting', *Journal of Experimental Psychology: Human Perception and Performance*, Vol. 8, No. 6, pp.778–792.

McGeoch, C. (1992) 'Analyzing algorithms by simulation: variance reduction techniques and simulation speedups', *ACM Computing Surveys*, Vol. 24, No. 2, pp.195–212.

Mehlhorn, K. and Marewski, J.N. (2011) 'Racing for the city: the recognition heuristic and compensatory alternatives', *Proceedings of the Interdisciplinary Workshop on Cognitive Neuroscience, Educational Research and Cognitive Modeling*, Delmenhorst, Germany.

Meyer, D.E. and Kieras, D.E. (1997) 'A computational theory of executive cognitive processes and multiple-task performance: Part 1. Basic mechanisms', *Psychological Review*, Vol. 104, No. 1, pp.3–65.

Newell, A. (1990) *Unified Theories of Cognition*, Harvard University Press, Cambridge, MA.

Newell, A. (Ed.) (1973) 'You can't play 20 questions with nature and win: projective comments on the papers of this symposium', Academic Press, New York.

Newell, A. and Rosenbloom, P.S. (1981) 'Mechanisms of skill acquisition and the law of practice', in J.R. Anderson (Ed.): *Cognitive Skills and their Acquisition*, pp.1–55, Erlbaum, Hillsdale, NJ.

Pavlik, P.I. and Anderson, J.R. (2005) 'Practice and forgetting effects on vocabulary memory: an activation-based model of the spacing effect', *Cognitive Science*, Vol. 29, No. 4, pp.559–586.

Rayner, K. (1998) 'Eye movements in reading and information processing: 20 years of research', *Psychological Bulletin*, Vol. 124, No. 3, pp.372–422.

Roland, P.E., Larsen, B., Lassen, N.A. and Skinhøj, E. (1980) 'Supplementary motor area and other cortical areas in organization of voluntary movements in man', *Journal of Neurophysiology*, Vol. 43, No. 1, pp.118–136.

Salthouse, T.A. (1984) 'Effects of age and skill in typing', *Journal of Experimental Psychology: General*, Vol. 113, No. 3, pp.345–371.

Salthouse, T.A. (1985) 'Anticipatory processing in transcription typing', *Journal of Applied Psychology*, Vol. 70, No. 2, pp.264–271.

Salthouse, T.A. (1986) 'Perceptual, cognitive, and motoric aspects of transcription typing', *Psychological Bulletin*, Vol. 99, No. 3, pp.303–319.

Salthouse, T.A. and Saults, J.S. (1987) 'Multiple spans in transcription typing', *Journal of Applied Psychology*, Vol. 72, No. 2, pp.187–196.

Salvucci, D.D. (2001) 'An integrated model of eye movements and visual encoding', *Cognitive Systems Research*, Vol. 1, No. 4, pp.201–220.

Salvucci, D.D. and Taatgen, N.A. (2008) 'Threaded cognition: an integrated theory of concurrent multitasking', *Psychological Review*, Vol. 115, No. 1, pp.101–130.

Schumacher, E.H., Seymour, T.L., Glass, J.M., Fencsik, D.E., Lauber, E.J., Kieras, D.E. et al. (2001) 'Virtually perfect time sharing in dual-task performance: uncorking the central cognitive bottleneck', *Psychological Science*, Vol. 12, No. 2, pp.101–108.

Squire, L.R. (2004) 'Memory systems of the brain: a brief history and current perspective', *Neurobiology of Learning and Memory*, Vol. 82, No. 3, pp.171–177.

Taatgen, N.A. and Anderson, J.R. (2002) 'Why do children learn to say "Broke"? A model of learning the past tense without feedback', *Cognition*, Vol. 86, No. 2, pp.123–155.

Wu, C. and Liu, Y. (2008a) 'Queuing network modeling of the psychological refractory period (PRP)', *Psychological Review*, Vol. 115, No. 4, pp.913–954.

Wu, C. and Liu, Y. (2008b) 'Queuing network modeling of transcription typing', *ACM Transactions on Computer-Human Interaction*, Vol. 15, No. 1, Article 6, pp.1–45.