

PENDEKATAN PERMODELAN MASA UNTUK ANALISIS PENJADUALAN DALAM PEMBANGUNAN PERISIAN BERASASKAN KOMPONEN

DAYANG NORHAYATI ABANG JAWAWI¹, RADZIAH MOHAMAD²,
ROSBİ MAMAT³, SAFAAI DERIS⁴ & MOHD ZULKIFLI MOHD ZAKI⁵

Abstrak. Dalam merekabentuk sesebuah perisian masa-nyata terbenam, perkara penting yang perlu dititikberatkan ialah mengambil kira keperluan masa-nyata perisian tersebut. Kebolehan meramal dan menganalisis masa boleh menjamin keutuhan sesebuah sistem masa-nyata terbenam. Keperluan pemasaan bagi sistem masa-nyata terbenam mestilah dimodelkan dengan jelas semasa fasa keperluan dan fasa reka bentuk. Ini bertujuan untuk mengelakkan ralat pemasaan berlaku semasa berada di lapangan dan menyebabkan kos yang banyak bagi kerja-semula pada peringkat akhir pembangunan. Analisis penjadualan merupakan alatan asas untuk menyemak ketepatan masa bagi sesebuah aplikasi masa-nyata. Kaedah analisis ini membolehkan proses menyemak faktor kekangan masa dengan meramal kes perlakuan terburuk bagi sesebuah sistem masa-nyata semasa algoritma penjadualan diaplikasikan. Kertas kerja ini mencadangkan satu pendekatan permodelan pemasaan di dalam pembangunan perisian yang berasaskan komponen dan menunjukkan bagaimana ramalan prestasi masa-nyata boleh dibuat berdasarkan model pemasaan tersebut. Satu eksperimen yang melibatkan kajian kes sistem masa-nyata terbenam telah direka bentuk untuk menunjukkan pendekatan yang dicadangkan ini dan juga mengesahkan keputusan ramalan prestasi sebenar perisian masa-nyata terbenam tersebut.

Kata kunci: Sistem masa-nyata terbenam; analisis penjadualan; perisian berasaskan komponen

Abstract. An important issue in designing embedded real-time (ERT) software is the consideration for real-time requirements of the software. The abilities to predict and analyze timing are key requirements for reliable ERT systems. The temporal requirements of the ERT system must be explicitly modeled during requirements and design phases to avoid timing error in the field and costly late rework. Scheduling analysis is a fundamental tool for checking timing correctness of a real-time application. It allows checking timing constraints by predicting the worst-case behavior of a real-time system when a scheduling algorithm is applied. The aims of this paper are to propose an approach which enables temporal modeling and to demonstrate prediction of real-time performance based on the temporal models in component-based software development. An experiment on case-study was designed, to demonstrate the approach and to validate the predicted results against the real performance of an ERT software.

Keywords: Embedded real-time systems; schedulability analysis; component-based software

^{1,2,4&5} Jabatan Kejuruteraan Perisian, Fakulti Sains Komputer dan Sistem Maklumat, Universiti Teknologi Malaysia 81310 UTM Skudai, Malaysia

³ Jabatan Kejuruteraan Mekatronik dan Robotik, Fakulti Kejuruteraan Elektrik, Universiti Teknologi Malaysia 81310 UTM, Skudai, Malaysia

Email: dayang@utm.my

1.0 PENGENALAN

Pembangunan perisian berasaskan komponen atau (*Component-Based Software Development, CBSD*) merupakan satu pendekatan yang semakin popular untuk meningkatkan produktiviti perisian, meningkatkan kualiti perisian dan pada masa yang sama mengurangkan kos pembangunan perisian. CBSD digunakan secara meluas dalam pelbagai aplikasi perisian seperti aplikasi kegunaan komputer peribadi, perisian berasaskan web teragih dan perniagaan elektronik atau e-perniagaan. Untuk menyesuaikan pendekatan CBSD kepada domain sistem masa-nyata terbenam adalah sukar terutamanya untuk sistem-sistem masa-nyata terbenam yang berskala kecil disebabkan oleh kekangan sumber dan keperluan masa-nyata sistem [1]. Beberapa kajian dan usaha industri telah berjaya mengaplikasikan CBSD kepada sistem masa-nyata terbenam dengan mengoptimumkan kelajuan dan penggunaan ingatan pada fasa reka bentuk semasa komposisi komponen dilakukan [1]–[4]. Pendekatan ini membolehkan proses mengoptimum secara global dibuat dan ramalan untuk keperluan sistem juga boleh dilakukan daripada ciri komponen yang diberikan. Namun begitu, banyak lagi kajian perlu dilakukan pada pendekatan-pendekatan ini bagi membolehkan teori dan kaedah yang sedia ada diaplikasikan dengan lebih berkesan kepada domain masa-nyata terbenam ini.

Algoritma analisis penjadualan yang sedia ada dalam pengkomputeran masa-nyata boleh digunakan secara praktikal dalam meramal prestasi sistem terbenam masa-nyata [2]. Bagi membolehkan analisis penjadualan dilakukan terhadap model reka bentuk berasaskan komponen atau model berorientasikan objek, pemetaan model kepada sifat-sifat pemasaan perlu dilakukan. Teknik pemetaan ini dipanggil sebagai strategi berbilang-tugasan. Di dalam CBSD, kebanyakan strategi berbilang-tugasan adalah berasaskan kepada model komponen. Strategi berbilang-tugasan yang berasaskan komponen ini meletakkan satu atau lebih komponen-komponen kepada satu tugas bergantung kepada pemilihan optimum ke atas komponen-komponen ini.

Terdapat dua strategi berbilang-tugasan yang boleh digunakan untuk model berorientasikan objek bagi *Unified Modeling Language for Real-Time (UML-RT)*. Dua strategi tersebut ialah berbilang-tugasan berasaskan kapsul [5–6] dan berbilang-tugasan berasaskan senario [7]. Berbilang-tugasan berasaskan kapsul adalah sesuai bagi CBSD kerana kapsul terdiri daripada objek boleh dipetakan secara terus kepada komponen, seperti yang telah dilakukan oleh Gu dan He [6]. Walau bagaimanapun, strategi berbilang-tugasan berasaskan kapsul ini tidak mematuhi andaian penjadualan Analisis Kadar Monotonik atau *Rate Monotonic Analysis (RMA)* kerana sebuah kapsul yang mengandungi sub-tugasan tidak boleh dipetakan secara terus kepada RMA.

RMA digunakan dengan meluas dalam pembangunan perisian masa-nyata kerana konsepnya mudah untuk dilaksanakan dan banyak piawai dan sistem pengendalian menyokong penggunaan penjadualan RMA.

Kertas kerja ini mencadangkan satu pendekatan bagaimana strategi berbilang-tugasan berasaskan kapsul boleh digunakan untuk menterjemahkan model analisis UML-RT kepada model reka bentuk komponen *Pervasive Component Systems* (PECOS) [4] bagi membolehkan analisis penjadualan RMA dilakukan terhadap model reka bentuk yang terhasil. Satu kajian kes sistem masa-nyata terbenam yang merupakan sistem robot berautonomi digunakan di dalam kertas kerja ini bagi menggambarkan pendekatan yang dicadangkan.

Tujuan utama perisian robot tersebut adalah untuk mengawal fungsi robot semasa bergerak di dalam persekitarannya. Sistem robot ini digerakkan sepasang roda. Setiap roda robot dipacu oleh motor arus terus. Kelajuan motor-motor robot akan diukur dengan menggunakan penderia halaju dan disuap balik kepada pengawal terbenam untuk menghasilkan isyarat kawalan kepada motor tersebut setiap beberapa milisaat dengan menggunakan algoritma kawalan pengkadar-kamiran (PI). Pengawal terbenam robot juga memantau persekitaran robot dengan menggunakan penderia kehampiran dan suis. Rajah 1 menunjukkan robot berautonomi yang digunakan sebagai kajian kes.



Rajah 1 Kajian kes robot berautonomi

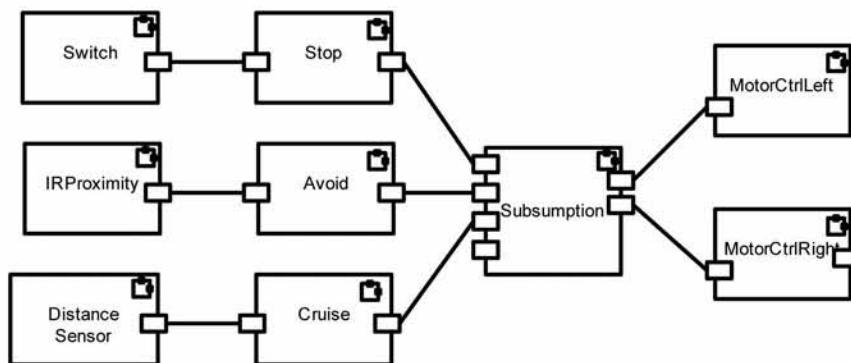
Kandungan kertas kerja ini disusun seperti berikut. Bahagian 2.0 dan Bahagian 3.0 membentangkan cadangan pendekatan permodelan pemasaan yang berasaskan strategi berbilang-tugasan menggunakan kapsul pada peringkat analisis dan peringkat reka bentuk. Pendekatan yang dicadangkan dijelaskan dengan menggunakan kajian

kes sistem robot di atas. Di Bahagian 4.0, ramalan perlakuan pemasaan sistem robot berasaskan RMA akan ditunjukkan. Keputusan eksperimen masa-nyata robot tersebut dan ramalan prestasi sistem robot menggunakan pendekatan yang dicadangkan dibincangkan secara terperinci dalam Bahagian 5.0. Kesimpulan bagi kertas kerja ini diberikan dalam Bahagian 6.0.

2.0 CIRI-CIRI PEMASAAN DALAM MODEL ANALISIS

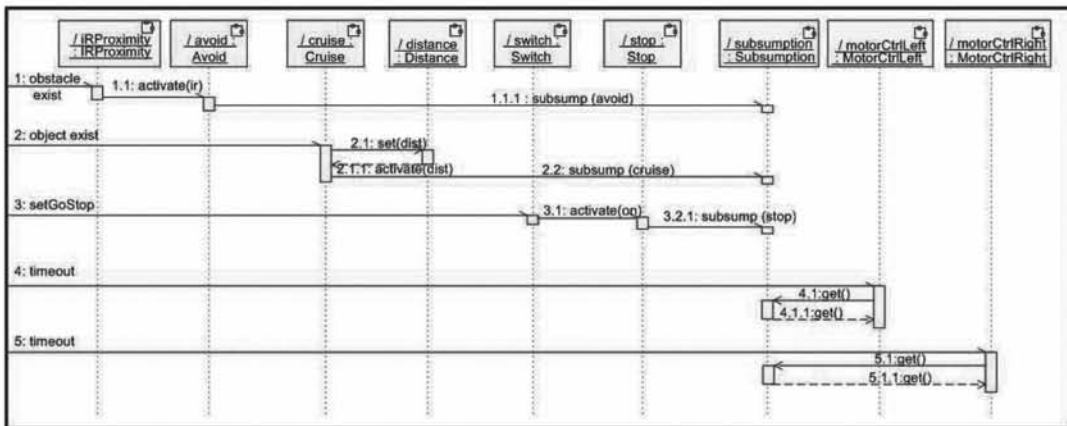
2.1 Pemodelan Aras Tinggi Komponen

Sistem masa-nyata terbenam memerlukan kaedah pemodulan seperti UML-RT untuk menyediakan model analisis yang mengambil kira spesifikasi pemasaan pada aktiviti kejuruteraan komponen. Permodelan UML-RT terdiri daripada permodelan struktur dan permodelan perlakuan. *Kapsul*, *pangkalan* dan *penyambung* merupakan tiga prinsip yang digunakan semasa proses membina struktur model dengan menggunakan UML-RT. *Kapsul* adalah kelas-kelas khusus yang mempunyai mekanisme komunikasi dan kumpulan objek yang sama sifatnya. *Pangkalan* pula merupakan objek yang bertindak sebagai antaramuka di lapisan sempadan kapsul. Manakala objek fizikal yang menghantar isyarat antara pangkalan-pangkalan adalah *penyambung*. Gambar rajah struktur untuk memodelkan perisian kajian kes robot berautonomi sebagai rangkaian kapsul ditunjukkan dalam Rajah 2. Setiap contoh kapsul mempunyai elemen kawalan logikal masing-masing dan berkomunikasi di antara satu sama lain dengan penghantaran dan penerimaan mesej melalui pangkalan. Di dalam pendekatan ini, kapsul dianggap sebagai komponen kerana kapsul boleh digunakan sebagai kumpulan objek kepada unit-unit operasi serba lengkap atau komponen, dan pangkalan boleh menyediakan mekanisme antaramuka antara kapsul.



Rajah 2 Model struktur UML-RT bagi sistem robot berautonomi

Protokol dan mesin keadaan merupakan dua binaan asas untuk permodelan perlakuan yang menggunakan UML-RT. Protokol ditakrifkan sebagai perjanjian berkontrak antara ahli-ahli yang berkomunikasi. Mesin keadaan pula memodelkan perlakuan kelas-kelas, protokol dan kapsul tertentu dalam memberikan tindak balas kepada mesej-mesej peristiwa. Di dalam sistem-sistem yang melibatkan masa-nyata, peristiwa-peristiwa dicetuskan secara dalaman antara sistem-sistem dan secara luaran daripada mesej luaran persekitaran sistem seperti pengesan, motor dan lain-lain. Peristiwa dan mesej perisian robot ini boleh dikenal pasti melalui gambar rajah jujukan seperti ditunjukkan dalam Rajah 3.



Rajah 3 Rajah jujukan menunjukkan jujukan mesej antara kapsul

Rajah 3 menunjukkan lima senario hujung-ke-hujung yang dicetuskan oleh mesej luaran. Mesej luaran 1 hingga 3 adalah dicetuskan oleh persekitaran yang mesejnya dibaca daripada pengesan-pengesan dan suis-suis. Mesej luaran 4 dan 5 pula dipandu oleh mesej masa tamat bagi kawalan gelung-tertutup motor robot. Kedua-dua permodelan struktur dan pemodelan perlakuan dalam Rajah 2 dan Rajah 3 digunakan untuk memodelkan niskala perisian robot berautonomi dan memudahkan proses pemahaman masalah. Ia juga memudahkan pemetaan keperluan sifat sistem robot berautonomi yang diperlukan untuk mengendali tindak balas dengan persekitarannya secara berterusan dan menepati masa yang ditentukan.

Dengan menggunakan model analisis ini, persekitaran fizikal perisian robot ditentukan pada aras tinggi sebagai satu struktur komponen-komponen dinamik yang berinteraksi. Model ini menyediakan analisis sistem yang berpandukan peristiwa dengan membayangkan bagaimana sistem masa-nyata sebenarnya berfungsi. Alatan

UML-RT dapat digunakan untuk mengesahkan keperluan perisian sistem yang melibatkan jurutera perisian dan pengguna berdasarkan kepada simulasi model analisis ini. Model UML-RT menyokong proses pengesahan keperluan kefungsiian dan keperluan perlakuan. Walau bagaimanapun, model pemasaan yang digunakan untuk mengesahkan kebenaran dan ketepatan pemasaan sistem-sistem masa-nyata terbenam semasa peringkat reka bentuk awalan tidak dinyatakan secara jelas di dalam alatan UML-RT. Oleh itu, penggunaan strategi berbilang-tugasan digunakan dalam kertas kerja ini untuk memperolehi pemboleh ubah pemasaan dari model UML-RT, supaya analisis penjadualan boleh dilakukan.

2.2 Mengenal Pasti Ciri-ciri Pemasaan

Di dalam strategi berbilang-tugasan berasaskan kapsul, terdapat dua jenis pendekatan yang boleh digunakan untuk menentukan keutamaan tugas, iaitu: penentuan keutamaan berasaskan senario dan penentuan keutamaan berasaskan kapsul. Berbilang-tugasan berasaskan kapsul dengan strategi penentuan keutamaan berasaskan kapsul telah dipilih di dalam pendekatan yang dicadangkan di sini. Di dalam strategi ini, satu atau lebih kapsul dikumpulkan kepada satu tugas dengan keseragaman keutamaan. Cara ini lebih sistematik jika digunakan di dalam CBSD kerana ia tidak memisahkan komponen yang telah ditentukan di sepanjang analisis model. Strategi yang sama telah digunakan oleh Gu dan He [6] dalam kerja mereka, tetapi mereka tidak menggunakan teori analisis penjadualan RMA kerana andaian pada teori RMA adalah tidak bersesuaian dengan jenis strategi mereka.

Di dalam kertas kerja ini, pendekatan yang dicadangkan membolehkan strategi yang telah dipilih menggunakan teori analisis penjadualan RMA dengan memetakan tugas-tugas daripada UML-RT kepada model komponen PECOS. Model PECOS akan menyediakan mekanisme untuk perkongsian data dan keserentakkan, serta dapat membolehkan andaian RMA memenuhi kesesuaian menggunakan berbilang-tugasan berasaskan kapsul dan strategi penentuan keutamaan berasaskan kapsul.

Berikut merupakan langkah-langkah atau peraturan-peraturan yang dicadangkan bersama dengan strategi berbilang-tugasan tersebut untuk diaplikasikan kepada domain robot berautonomi:

- (1) Mana-mana kapsul yang diakses oleh berbilang senario dianggap sebagai satu tugas.
- (2) Mana-mana kapsul yang hanya diakses oleh satu senario boleh dikumpulkan sebagai satu tugas bersama-sama dengan kapsul lain (hanya diakses oleh senario yang sama) yang berinteraksi dengan senario tersebut.

- (3) Tugas-tugas yang dicetuskan oleh mesej-mesej luaran akan ditentukan masa kitaran berkala secara eksperimen.
- (4) Setiap tugas yang mempunyai tempoh ditentukan keutamaan berdasarkan kepada algorithma penjadualan RMA yang menentukan keutamaan tugas berkala secara songsang kepada tempoh tugas.
- (5) Tugas yang tidak dicetus oleh mesej-mesej luaran akan ditentukan keutamaan yang mana keutamaannya adalah satu peringkat lebih rendah daripada keutamaan tugas tertinggi yang berinteraksi dengan kapsul tersebut. Tempohnya boleh disetkan kepada sebarang nombor di antara tempoh keutamaan paling tinggi dan tempoh untuk keutamaan tugas tertinggi seterusnya.

Bagi mendemostrasikan proses berbilang-tugas yang dicadangkan di atas, pertimbangkan rajah jujukan di dalam Rajah 3. Daripada rajah jujukan dan langkah yang dinyatakan di atas, terdapat enam tugas yang telah dikenal pasti semasa proses analisis dilakukan terhadap jujukan peristiwa dan mesej. Salah satu tugas yang telah dikenal pasti daripada satu kapsul yang telah diakses oleh berbilang senario (daripada Langkah 1) ialah tugas yang terdiri daripada kapsul subsumption. Lima tugas telah dikenal pasti daripada kapsul-kapsul yang hanya boleh diakses oleh satu senario (daripada Langkah 2) adalah: satu tugas yang terdiri daripada IPProximity dan avoid yang dipanggil sebagai tugas Avoid; satu tugas terdiri daripada distanceSensor dan cruise yang dipanggil sebagai tugas Cruise; satu tugas yang terdiri daripada switch dan stop yang dipanggil sebagai tugas Stop; satu tugas yang terdiri daripada motorCtrlRight; dan satu tugas yang terdiri daripada motorCtrlLeft.

Tugas-tugas yang telah dicetuskan oleh mesej-mesej luaran adalah tugas Motorctrl_left, Motorctrl_right, Stop, Avoid dan Cruise dan telah diumpukkan masa kitaran berkala yang ditentukan berdasarkan kepada kekerapan mesej luaran dan eksperimen (daripada Langkah 3). Daripada eksperimen kawalan motor yang dijalankan, didapati tempoh untuk tugas Motorctrl_left dan Motorctrl_right telah disetkan kepada 20 milisaat dan daripada ramalan yang dilakukan untuk masa ketibaan mesej luaran terburuk, tempoh untuk tugas Stop telah disetkan kepada 30 milisaat, tugas Avoid telah disetkan kepada 50 milisaat dan tugas Cruise telah disetkan kepada 50 milisaat. Tugas Subsumption adalah tugas yang tidak dicetuskan oleh mesej-mesej luaran (daripada Langkah 5) dan keutamaan tugas telah ditentukan kepada satu peringkat lebih rendah daripada keutamaan tugas tertinggi yang berinteraksi dengan tugas Subsumption (satu peringkat lebih rendah

daripada keutamaan untuk tugas Motorctrl_left dan Motorctrl_right). Jadual 1 menunjukkan senarai untuk kesemua tugas-tugas dan tempoh-tempoh dan juga penentuan keutamaan berdasarkan kepada langkah-langkah yang dicadangkan.

Jadual 1 Tugas-tugas robot dan ciri-ciri pemasannya

Elemen	Tempoh (msaat)	Keutamaan
Motorctrl_left	20	4 (tertinggi)
Motorctrl_right	20	4
Subsumption	30	3
Stop	30	2
Avoid	50	1
Cruise	50	1(terendah)

3.0 CIRI-CIRI PEMASAAN DI DALAM MODEL REKA BENTUK

Model analisis yang dibincangkan dalam bahagian sebelum ini perlu diterjemahkan kepada reka bentuk dan implementasi yang lebih konkrit. Satu model komponen diperlukan untuk memperincikan piawaian dan peraturan yang perlu dipatuhi oleh pembangun komponen dalam aktiviti kejuruteraan komponen dan pembangun aplikasi yang mengguna-semula komponen tersebut dalam aktiviti CBSD. Salah satu model komponen yang dibangunkan untuk sistem terbenam masa-nyata ialah PECOS. Model PECOS menyediakan mekanisme untuk perkongsian data dan pengurusan keserentakkan bagi himpunan komponen-komponen. Langkah-langkah utama yang terlibat di dalam melaksanakan perisian masa-nyata terbenam menggunakan model komponen PECOS adalah pengenalpastian komponen-komponen yang diperlukan daripada model keperluan aplikasi, komposisi komponen-komponen, dan menentukan tempoh dan keutamaan bagi komponen-komponen. Untuk melaksanakan langkah-langkah ini ke atas kajian kes yang telah dipilih, satu transformasi yang bersistematik daripada model berasaskan objek UML-RT yang telah dihasilkan dari pemodelan di Bahagian 2.0 kepada model komponen PECOS diperlukan. Proses transformasi ini terdiri daripada mengenal pasti komponen-komponen dan perkongsian data antara komponen-komponen.

Dalam proses transformasi ini, kapsul-kapsul dianggap sebagai komponen-komponen dan pangkalan menyediakan mekanisme antaramuka untuk kapsul. Oleh itu, model struktur UML-RT bagi sistem robot berautonomi seperti dalam Rajah 2 dapat ditukarkan terus kepada model struktur PECOS. Namun begitu, perbezaan

cara komunikasi antara kapsul atau komponen yang digunakan dalam UML-RT (berorientasikan mesej) dan PECOS (berorientasikan pengaliran data) merupakan satu isu yang menghalang pemetaan terus antaramuka komponen antara dua model ini. Ini memerlukan kepada proses pemetaan untuk jenis antaramuka komponen yang berbeza-beza yang melibatkan elemen pangkalan dan protokol dalam model UML-RT dan elemen pangkalan dalam model PECOS.

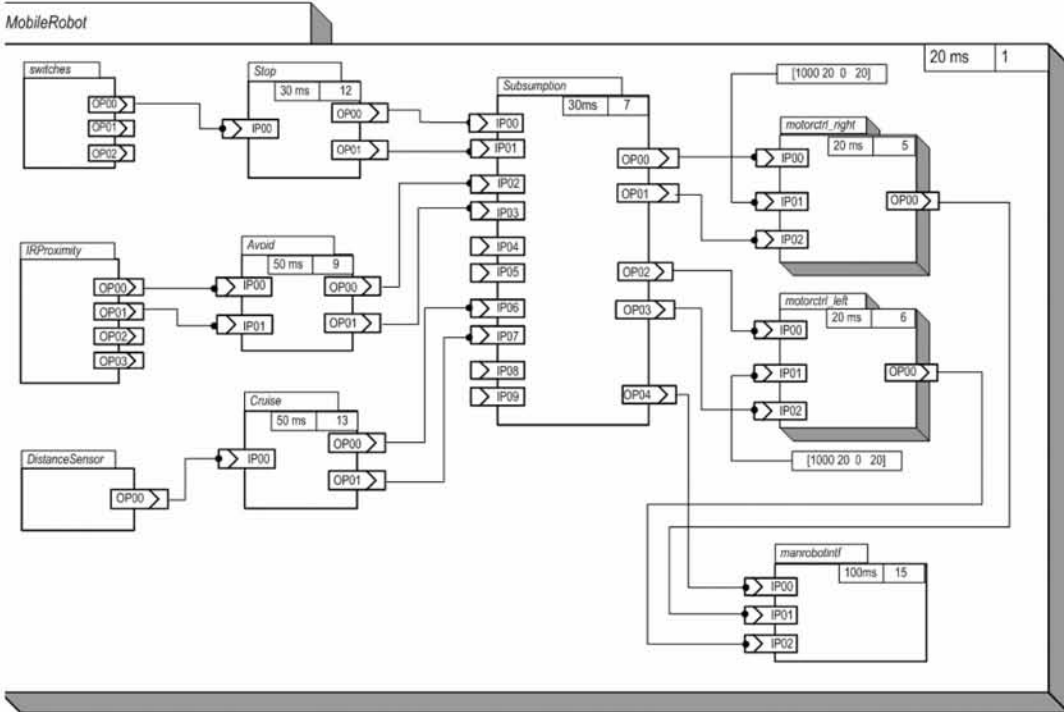
Perkongsian data antara kapsul dapat dikenal pasti dengan mengesan mekanisme komunikasi dalam UML-RT (protokol dan pangkalan) yang menghubungkan dua kapsul. Apabila dua kapsul ini berkomunikasi, mesej akan dihantar di antara mereka. Mesej boleh digunakan untuk menyampaikan maklumat atau nilai data daripada satu kapsul kepada yang lain. Maklumat dan nilai data tersebut diperlakukan sebagai pangkalan dalam model antaramuka PECOS. Dengan cara ini pemetaan antaramuka model UML-RT dan PECOS boleh dilakukan.

Hasil yang diperolehi daripada proses penukaran model berorientasikan objek UML-RT kepada model komponen PECOS merupakan rajah komposisi komponen seperti yang ditunjukkan dalam Rajah 4. Komposisi ini terdiri daripada lapan komponen aktif, tiga komponen pasif dan tiga komponen komposit. Di dalam rajah ini, komponen aktif ialah komponen-komponen yang mempunyai nilai tempoh dan keutamaan dan komponen-komponen tanpa nilai-nilai ini adalah komponen pasif. Di dalam komposisi ini juga, komponen `manrobotintf` telah ditambahkan untuk tujuan pengawasan operator supaya robot dapat berkomunikasi dengan manusia menggunakan paparan cecair kristal dengan memaparkan taraf robot, perlakuan dan kelajuan semasa robot bergerak. Bahagian yang seterusnya akan membincangkan bagaimana analisis penjadualan RMA boleh digunakan untuk menganalisis ciri-ciri pemasaan model reka bentuk komponen yang diperolehi dari strategi yang telah dicadangkan.

4.0 PENJADUALAN MENGGUNAKAN ANALISIS PERKADARAN MONOTONIK

Lui dan Layland [8] telah membangunkan Analisis Perkadaran Monotonik (RMA) yang dihasil dalam tahun 1970an. Ia telah dibangunkan bermula daripada model yang sangat ringkas di mana semua tugas adalah berkala dan bebas. RMA yang ringkas di dalam Teorem 1 adalah untuk menyemak set tugas bebas dan penjadualan yang berkala.

Teorem 1: Satu set dengan n tugas berkala bebas boleh dijadualkan menggunakan algoritma perkadaran monotonik, jika



Rajah 4 Komposisi komponen-komponen robot berautonomi menggunakan model PECOS

$$\sum_{i=1}^n \frac{C_i}{T_i} \leq U_n = n \left(2^{\frac{1}{n}} - 1 \right), n = 1, 2, \dots \tag{1}$$

di mana C_i ialah masa komputasi tugas yang ke- i , T_i ialah tempoh tugas yang ke- i dan U_n ialah had penggunaan bagi n tugas.

Kes terburuk had penggunaan dalam teorem ini dianggapkan ketat kerana wujud set-set tugas yang tidak dapat dilaksanakan jika penggunaan arbitari mendekati nilai $n (2^{1/n} - 1)$. Lehoczky telah menunjukkan bahawa purata penggunaan penjadualan, bagi set tugas besar yang dipilih secara rawak, dapat meningkat daripada julat 88% sehingga 95% [9]. Penggunaan yang tinggi ini boleh dijamin dengan pemilihan tempoh tugas yang betul. Algoritma penjadualan RMA dapat menjadualkan tugas sehingga 100% penggunaan bagi tugas adalah $D_i = T_i$ dan jika tempoh tugas adalah selaras, di mana setiap tempoh tugas adalah nilai tepat gandaan integer tepat tempoh tugas yang seterusnya [9].

Semasa mereka bentuk perisian untuk sistem robot berautonomi yang menggunakan kawalan kepintaran berasaskan perlakuan, salah satu persoalan yang sering ditimbulkan ialah “berapakah jumlah maksimum perlakuan yang boleh dilaksanakan oleh robot dan pada masa yang sama memenuhi keperluan masa?” Persoalan ini boleh dijawab dengan meramalkan had teratas terhadap jumlah perlakuan yang boleh dilaksanakan menggunakan analisis RMA. Untuk membantu proses ramalan ini, beberapa andaian telah dibuat:

- (1) Seni bina perisian mestilah mengikut seni bina lapisan perlakuan yang terdiri daripada satu set pengesan, perlakuan, penggerak dan satu unit penimbangtara (*subsumption*).
- (2) Implementasi dan komposisi komponen mestilah mengikut model PECOS. Rajah 4 menunjukkan contoh komposisi robot berautonomi.

Tugasan-tugasan di dalam sistem robot berautonomi boleh dibahagikan kepada tiga kumpulan berdasarkan kepada keperluan masa-nyata: keperluan masa-nyata keras, keperluan masa-nyata kukuh, dan keperluan masa-nyata lembut. Set tugas masa-nyata keras mewakili proses-proses berkala yang tidak bergantung kepada aplikasi robot. Contohnya seperti proses-proses pengawal motor gelung tertutup di mana sebarang proses yang gagal boleh melibatkan faktor keselamatan robot atau manusia. Di dalam Rajah 4, komponen-komponen `motorctrl_right`, `motorctrl_left` dan `MobileRobot` dianggap sebagai tugas masa-nyata keras. Set tugas masa-nyata kukuh pula menyokong kepada pelaksanaan proses perlakuan yang menentukan kereaktifan robot. Di dalam Rajah 4, komponen-komponen yang dianggap sebagai tugas masa-nyata kukuh ialah komponen `Aviod`, `Cruise`, `Stop` dan `Subsumption`. Set tugas ketiga ialah masa-nyata lembut, yang merujuk kepada proses-proses yang membenarkan perwakilan lapisan pertengahan dan lapisan perundingan. Komponen `manrobotintf` dianggap sebagai tugas masa-nyata lembut.

Daripada Teorem 1, pengujian penjadualan bagi tugas-tugas dengan RMA menakrifkan penggunaan U_n . Disebabkan bilangan tugas dan masa komputasi untuk tugas-tugas ini bergantung kepada bilangan pengesan, penggerak dan perlakuan yang ada pada seni bina lapisan perlakuan robot, beberapa andaian telah dibuat untuk memudahkan ramalan had atas bagi jumlah bilangan perlakuan:

- (1) Terdapat dua kumpulan untuk tempoh tugas, iaitu: T_b ialah tempoh tugas bagi tugas masa-nyata kukuh dan T_a ialah tempoh tugas bagi tugas masa-nyata keras.

- (2) Semua pengesan, penggerak dan komponen induk penggerak dimiliki oleh tugas masa-nyata keras; sementara penimbangtara dan semua komponen perlakuan dimiliki oleh tugas masa-nyata kukuh.

Oleh itu, penggunaan sistem robot berautonomi ialah satu fungsi jumlah bilangan perlakuan robot N_b , jumlah bilangan penggerak N_a dan jumlah bilangan pengesan N_s adalah:

$$\begin{aligned}
 U(N_b, N_a, N_s) &= \sum_{i=1}^n \frac{C_i}{T_i} \\
 &= \sum_{i=1}^{N_b} \frac{C_{b_i}}{T_b} + \sum_{i=1}^{N_a} \frac{C_{a_i}}{T_a} + \sum_{i=1}^{N_s} \frac{C_{s_i}}{T_a} + k
 \end{aligned} \tag{2}$$

di mana C_{b_i} ialah masa komputasi komponen perlakuan yang ke- i , C_{a_i} ialah masa komputasi komponen penggerak yang ke- i , C_{s_i} ialah masa komputasi komponen pengesan yang ke- i , T_a ialah tempoh tugas bagi tugas masa-nyata keras, T_b ialah tempoh tugas bagi tugas masa-nyata kukuh, dan k ialah jumlah penggunaan bagi komponen subsumption dan komponen induk yang mana merupakan pemalar kepada implementasi yang dinyatakan. Daripada (2), didapati bahawa, dengan mengetahui nilai-nilai bagi N_a , N_s dan k , satu anggaran had atas bagi jumlah bilangan perlakuan robot N_b dapat diramalkan.

5.0 HASIL EKSPERIMEN

Untuk menilai pendekatan permodelan yang dicadangkan dalam meramal perlakuan pemasaan robot berautonomi ini, satu eksperimen melibatkan proses pelaksanaan masa-nyata robot telah dibuat. Proses ramalan yang dilakukan menggunakan pendekatan yang dicadangkan ini ditentukan secara eksperimen terhadap sistem robot yang telah dilaksanakan. Di dalam eksperimen ini, keutuhan perlakuan dan kereaktifan robot akan dianalisis semasa proses pelaksanaan robot yang sebenar berdasarkan ciri-ciri pemasaan yang diterbitkan daripada ramalan di atas. Permodelan robot berautonomi yang menggunakan perisian berasaskan komponen seperti yang ditunjukkan pada Rajah 4 telah ditulis untuk pengawal mikro 16-bit AMD188ES dengan kapasiti ingatan 64Kb ROM dan 128Kb RAM.

Alatan perisian yang digunakan untuk proses pembangunan perisian ialah pengkompil Paradigm C [10] untuk menjana kod yang boleh diaturcara ke dalam ROM dan kernel masa-nyata $\mu\text{C}/\text{OS-II}$ [11] untuk sokongan penugasan berbilang.

Berasaskan kepada pengukuran masa komputasi sistem robot semasa larian, telah didapati bahawa: setiap perlakuan dan komponen-komponen prajangka mempunyai masa komputasi satu milisaat, contohnya untuk semua i , $C_{bi} = 1$, dan masa komputasi untuk penggerak (C_a) ialah 3 milisaat setiap satu. Pada robot ini, terdapat dua penggerak ($N_a = 2$), oleh itu proses penggantian nilai-nilai $C_{bi} = 1$, C_a dan N_a ke dalam (1) menghasilkan:

$$U(N_b, 2, N_s) = \frac{N_b}{T_b} + \frac{6}{T_a} + \sum_{i=1}^{N_s} \frac{C_{bi}}{T_a} + k. \quad (3)$$

Oleh kerana setiap perlakuan komponen adalah dikaitkan dengan satu komponen pengesan, N_s adalah bersamaan dengan N_b . Oleh itu, (3) diringkaskan kepada:

$$U(N_b, 2) = \frac{N_b}{T_b} + \frac{6 + N_b}{T_a} + k. \quad (4)$$

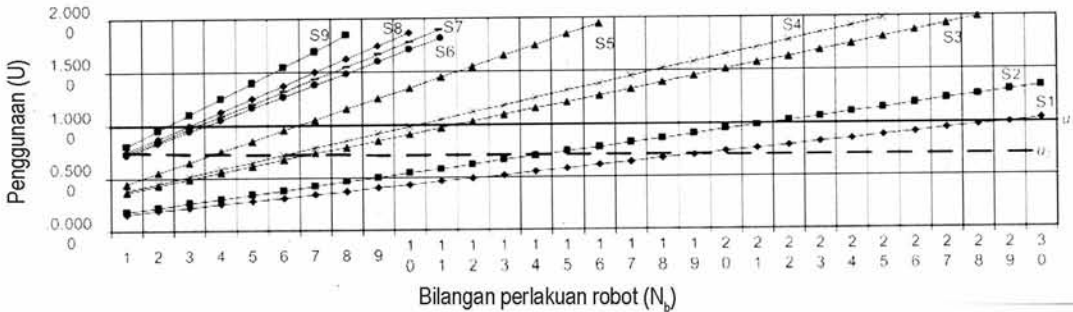
Fungsi $U(N_b, 2)$ dianalisis untuk mendapatkan nilai-nilai T_a dan T_b yang berlainan dan munasabah untuk sistem robot tersebut. Nilai-nilai ini adalah dipilih berdasarkan kepada keperluan kawalan gelung tertutup motor-motor dalam sistem robot berautonomi.

Sembilan set nilai bagi T_a dan T_b yang akan dianalisis dalam eksperimen ini disenaraikan dalam Jadual 2. Rajah 5 menunjukkan fungsi penggunaan $U(N_b, 2)$ dengan sembilan set nilai tetapan bagi T_a dan T_b tersebut. Lengkung penggunaan dilabelkan dengan tetapan yang berbeza daripada S1 hingga S9. Rajah 5 menunjukkan jumlah maksimum untuk perlakuan yang dibenarkan (di mana nilai penggunaan adalah di bawah 1) dan ditentukan oleh pemilihan T_a dan T_b .

Jadual 2 Variasi tetapan bagi T_a dan T_b

Tetapan	T_a	T_b
S1	50	100
S2	50	50
S3	20	100
S4	20	60
S5	20	20
S6	10	100
S7	10	60
S8	10	40
S9	10	20

Di dalam Rajah 5, dua garisan yang mewakili sempadan penggunaan diplotkan untuk mengenal pasti syarat cukup untuk sempadan atas bagi jumlah perlakuan robot. Had penggunaan pertama adalah berdasarkan kepada Lehoczky [9], iaitu 100% penggunaan yang diwakili oleh garisan lurus berwarna hitam gelap apabila $u_1 = 1$ dan had penggunaan kedua, berdasarkan kepada Liu dan Layland [8], iaitu $u_2 = n(2^{1/n} - 1)$ yang diwakili oleh garisan putus-putus berwarna gelap. Penggunaan bersamaan dengan 1 boleh digunakan dalam eksperimen ini disebabkan $D_i = T_i$ dan tempoh setiap tugas adalah selaras. Walau bagaimanapun, set tugas, batasan waktu dan tempoh tugas untuk proses implementasi robot ini mungkin akan berubah, maka syarat umum penjadualan u_2 adalah lebih sesuai digunakan jika perubahan ini dibuat.



Rajah 5 Penyelesaian untuk jumlah maksimum bagi N_b untuk kajian kes robot berautonomi berdasarkan kepada penggunaan U dan 100% untuk sembilan variasi tetapan bagi T_a dan T_b

Berdasarkan kepada nilai-nilai T_a yang ditunjukkan dalam Rajah 5, persilangan untuk setiap nilai dengan $u_2 = n(2^{1/n} - 1)$ dan $u_1 = 1$ diperhatikan. Persilangan ini boleh diperhatikan dalam tiga kumpulan berdasarkan nilai T_a : S1 dan S2 bagi $T_a = 50$, S3–S5 bagi $T_a = 20$ dan S6–S9 bagi $T_a = 10$. Jadual 3 menyenaraikan titik persilangan minimum dengan $u_2 = n(2^{1/n} - 1)$ dan titik persilangan maksimum dengan $u_1 = 1$ bagi setiap kumpulan T_a .

Hasil yang diperolehi daripada Jadual 3 boleh digunakan dalam menganggar jumlah maksimum perlakuan yang boleh dilaksanakan oleh sistem robot berautonomi ini. Contohnya, dengan $T_a = 50$, menggunakan teorem RMA, robot berautonomi yang digunakan dalam eksperimen ini mampu untuk mengawal perlakuan sehingga 14 bilangan perlakuan robot. Jika syarat $D_i = T_i$ dan tempoh tugas adalah selaras dipatuhi, jumlah perlakuan boleh meningkat sehingga 29.

Jadual 3 Titik persilangan bagi kumpulan T_a dengan u_1 dan u_2

T_a	$n(2^{1/n} - 1)$	1
50	14	29
20	4	11
10	1	3

Menggunakan komposisi yang ditunjukkan dalam Rajah 4, satu set ciri pemasaan yang dicadangkan daripada Jadual 3 dilaksanakan bersama dengan tiga perlakuan ($N_b = 3$). Ciri-ciri pemasaan untuk komponen-komponen MobileRobot dijadualkan ke dalam Jadual 4 dengan senarai komponen aktif dan tempoh untuk setiap komponen. Setiap tugas komponen dalam jadual disusun mengikut keutamaan yang paling tinggi hingga keutamaan yang terendah.

Hasil eksperimen yang dijalankan menunjukkan bahawa perisian robot berautonomi ini adalah mempunyai sifat keutuhan perisian dan bertindak-balas secara aktif terhadap persekitarannya. Daripada keputusan ini, boleh dirumuskan bahawa ujian penjadualan had penggunaan RMA terhadap sistem terbenam masa-nyata ini membolehkan ramalan prestasi masa-nyata dibuat untuk memastikan pelaksanaan perlakuan perisian yang stabil dan memenuhi keperluan-keperluan fungsi yang telah ditetapkan.

Jadual 4 Ciri-ciri pemasaan MobileRobot's

Tugasan Komponen	Tempoh Tugas (ms)
MobileRobot	10
Motorctrl_left	10
Motorctrl_right	10
Subsumption	20
Stop	20
Avoid	20
Cruise	20
Manrobotintf	100

6.0 KESIMPULAN

Kertas kerja ini membincangkan, proses permodelan pemasaan dicadangkan untuk pembangunan perisian berasaskan komponen yang disokong oleh alatan UML-RT. Di dalam proses ini, isu utama yang diselesaikan ialah proses pemetaan model keperluan dan mereka bentuk model-model kepada tugas dan ciri pemasaannya. Kaedah penyelesaian yang dicadangkan dalam kertas kerja ini adalah merupakan pemetaan tahap tinggi yang boleh mengenal pasti ciri-ciri pemasaan pada fasa keperluan yang menggunakan model UML-RT bersama strategi berbilang-tugas berasaskan kapsul dan fasa reka bentuk yang menggunakan model berorientasikan komponen PECOS.

RMA merupakan analisis penjadualan yang membolehkan ramalan dibuat ke atas model pemasaan dan model reka bentuk berorientasikan komponen bagi sistem masa-nyata terbenam di dalam kertas kerja ini. Pendekatan permodelan dan penjadualan yang dicadangkan digambarkan dengan eksperimen masa-nyata yang dijalankan pada kajian kes sistem robot berautonomi. Berdasarkan kepada analisis dan eksperimen yang telah dijalankan, penjadualan RMA digunakan untuk mengenalpasti jumlah perlakuan yang boleh dilaksanakan di dalam perisian robot berautonomi berasaskan pendekatan yang dicadangkan. Pendekatan yang dicadangkan ini secara amnya membolehkan jurutera robotik mengesahkan reka bentuk yang telah dihasilkan memenuhi ketepatan masa pada aras tinggi pembangunan perisian dengan menggunakan kaedah kejuruteraan perisian dan teknologi masa-nyata.

Gherbi dan Khendek dalam [12] juga mencadangkan kaedah permodelan masa untuk model analisa penjadualan dengan menggunakan teknik senibina pacuan model (MDA) dan UML profil. Kaedah yang dicadangkan dalam kertas kerja ini lebih general dan mudah untuk diaplikasikan dengan cara manual berbandingkan dengan cadangan Gherbi dan Khendek lebih terperinci yang lebih mudah untuk diautomasikan ke dalam perkakasan perisian.

RUJUKAN

- [1] Crnkovic, I. 2004. Component-Based Approach for Embedded Systems. *Proc. 9th International Workshop on Computer Oriented Programming*. Oslo.
- [2] Wall, A. 2003. Architectural Modeling and Analysis of Complex Real-Time Systems: Ph.D. Thesis, Mälardalen Research and Technology Centre, Mälardalen University, Sweden.
- [3] Ommering, R. Linden, F. Kramer, J. dan Magee, J. 2000. *The Koala Component Model for Consumer Electronics Software*. *IEEE Computer*. 33(3): 78–85.
- [4] Nierstrasz, O. Arévalo, G. Ducasse, S. Wuyts, R. Black, A. Müller, P. Zeidler, C. Genssler, dan T. van den Born. R. 2002. A Component Model for Field Devices. *Proc. First International IFIP/ACM Working Conf. on Component Deployment*. Springer-Verlag Heidelberg, Berlin: 200–209.

- [5] Saksena, M. dan Karvelas, P. 2000. Designing for Schedulability: Integrating Schedulability Analysis with Object-Oriented Design. *IEEE Proceeding of EuroMicro Conference on Real-Time Systems*. Jun 19–21. Stockholm, Sweden: 101–108.
- [6] Gu, Z. dan He, Z. 2005. Real-Time Scheduling Techniques for Implementation Synthesis from Component-Based Software Models. Heineman, G. T., Crnkovic, I., Schmidt, H. W., Stafford, J. A., Szyperski, C. dan Wallnau, K. *Component-Based Software Engineering 2005, Lecture Notes in Computer Science 3489*. Berlin, Heidelberg:Springer-Verlag: 235–250.
- [7] Masse, J. Kim, S. dan Hong, S. 2003. Tool Set Implementation for Scenario-based Multithreading of UML-RT Models and Experimental Validation. *IEEE Proceeding of Real-Time and Embedded Technology and Applications Symposium*. May 27-30. Toronto, Canada: 70–77.
- [8] Lui C. L. dan Layland, J. 1973. Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment. *Journal of the ACM*. 20(1): 46–61.
- [9] Lehoczky, J. P. 1990. Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadlines. *Proceedings of 11th Real-Time Systems Symposium*. 5-7 Dis: 201–209.
- [10] *Paradigm C++ Reference Manual Version 5.0*, 2000. Paradigm Systems, Endwel.
- [11] Labrosse, J. J. 1999. *MicroC/OS-II The Real-Time Kernel*. 2nd Edition, Berkeley: R&D Books.
- [12] Gherbi A. dan Khendek F. 2009. From UML/SPT Model to Schedulability Analysis: Approach and a Prototype Implementation using ATL. *Automated Software Engineering*. 10.1007/s10515-008-0045-5.