【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第３区分
【発行日】平成27年8月13日(2015.8.13)

【公表番号】特表2014-526740(P2014-526740A)
【公表日】平成26年10月6日(2014.10.6)
【年通号数】公開・登録公報2014-055
【出願番号】特願2014-529889(P2014-529889)
【国際特許分類】
Ｇ０６Ｆ　9/54　(2006.01)
【ＦＩ】
Ｇ０６Ｆ　9/46　４８０Ｄ

【手続補正書】
【提出日】平成27年6月19日(2015.6.19)
【手続補正１】
【補正対象書類名】特許請求の範囲
【補正対象項目名】全文
【補正方法】変更
【補正の内容】
【特許請求の範囲】
【請求項１】
ミドルウェアまたはその他の環境において使用される動的呼出しおよびサービスインターフェイスを提供するためのシステムであって、
クライアントコンピュータ、クライアントコンテナ、およびクライアントアプリケーションを含むクライアントサイド環境と、
サービスプロバイダコンピュータ、サービスプロバイダコンテナ、およびサービスを含むサービスサイド環境と、
クライアントサイドおよびサービスサイドで動作可能な動的呼出しおよびサービスインターフェイスとを備え、
前記クライアントサイドにおいて、
前記クライアントアプリケーションがメッセージを処理のためにランタイムに置くことができるようにするディスパッチャリクエスト関数が与えられ、かつ、前記クライアントアプリケーションがメッセージを処理後にランタイムから受けることができるようにするディスパッチャレスポンス関数が与えられ、
メッセージ処理をトランスポートから切離すクライアントリクエストトランスポート関数およびクライアントレスポンストランスポート関数が与えられ、
前記サービスサイドにおいて、
メッセージング処理をトランスポートから切離すサービスリクエストトランスポート関数およびサービスレスポンストランスポート関数が与えられ、
前記サービスがメッセージを前記ランタイムを介して受けて処理することができるようにするプロバイダリクエスト関数と、前記サービスがレスポンスメッセージを処理のためにランタイムに置くことができるようにするプロバイダレスポンス関数とが与えられる、システム。
【請求項２】
各リクエスト関数は、前記クライアントサイドおよび前記サービスサイドにおいて、対応するコールバック関数を有し、それによりメッセージ処理を非同期にする、請求項１に記載のシステム。
【請求項３】
前記サービスは、ウェブサービスであり、

前記サービスプロバイダコンテナは、ウェブサービスコンテナであり、
前記クライアントアプリケーションは、ウェブサービスクライアントである、請求項１または２に記載のシステム。
【請求項４】
前記クライアントサイドのアウトバウンドにおいて、
前記クライアントアプリケーションは、前記ディスパッチャリクエスト関数をコールすることによって、ターゲットサービスに対するリクエストを行ない、
前記リクエストは、アウトバウンドのクライアントサイドＳＯＡＰスタックによって処理され、
前記ＳＯＡＰスタックはクライアントリクエストトランスポート関数をコールし、
前記クライアントコンテナにおける前記クライアントリクエストトランスポート関数は、前記リクエストをトランスポートに伝え、
前記サービスサイドのインバウンドにおいて、
前記サービスリクエストトランスポート関数は前記リクエストを前記トランスポートから受け、
前記サービスリクエストトランスポート関数は前記リクエストを処理のためにインバウンドのＳＯＡＰスタックに伝え、
前記ＳＯＡＰスタックはプロバイダリクエスト関数を前記サービスによる処理のためにコールし、
前記サービスサイドのアウトバウンドにおいて、
前記サービスがレスポンスを与えることができる状態になると、前記サービスはプロバイダレスポンス関数をコールして前記レスポンスを処理のためにアウトバウンドのサービスサイドＳＯＡＰスタックに伝え、
前記ＳＯＡＰスタックは前記サービスレスポンストランスポート関数をコールし、
前記サービスレスポンストランスポート関数は、前記レスポンスを前記トランスポートに伝え、
前記クライアントサイドのインバウンドにおいて、
前記クライアントコンテナは前記レスポンスを前記トランスポートから受け、
前記クライアントコンテナはクライアントレスポンストランスポート関数をコールして前記レスポンスをインバウンドのクライアントサイドＳＯＡＰスタックに伝え、
ＳＯＡＰ処理後、前記ＳＯＡＰスタックは、前記リクエストを前記クライアントアプリケーションに送るために前記ディスパッチャレスポンス関数に伝える、請求項１から３のいずれかに記載のシステム。
【請求項５】
リクエストヘッダに含まれるレスポンスアドレスに応じて前記レスポンスを前記クライアントアプリケーションに伝えるために、２つの異なるサービスレスポンストランスポート関数を与え選択的に呼出すことができ、
前記クライアントサイドにおいて、
前記クライアントアプリケーションがメッセージを処理のためにアウトバウンドのランタイムプロトコルスタックに非同期で置くことができるようにするディスパッチャリクエスト関数が与えられ、
処理後にレスポンスをインバウンドのプロトコルスタックから非同期で受けることができるようにするディスパッチャレスポンス関数が与えられ、
メッセージ処理をトランスポートから切離し、かつメッセージを非同期でハンドリングできるようにする、クライアントリクエストトランスポート関数およびクライアントレスポンストランスポート関数が与えられ、
前記サービスサイドにおいて、
メッセージング処理をトランスポートから非同期で切離し、かつプロトコルスタックがメッセージを前記ランタイムを介して受けて処理できるようにする、サービスリクエストトランスポート関数およびサービスレスポンストランスポート関数が与えられ、

前記プロトコルスタックによるインバウンド処理後にリクエストメッセージを非同期で受けるプロバイダリクエスト関数が与えられ、
サービスレスポンスをアウトバウンド処理のためにランタイムプロトコルスタックに非同期で置くことができるようにするプロバイダレスポンス関数が与えられる、請求項４に記載のシステム。
【請求項６】
ミドルウェアまたはその他の環境において使用されるメッセージングアプリケーションプログラミングインターフェイス（ＡＰＩ）を提供するための方法であって、
クライアントコンピュータ、クライアントコンテナ、およびクライアントアプリケーションを含むクライアントサイド環境を提供するステップと、
サービスプロバイダコンピュータ、サービスプロバイダコンテナ、およびサービスを含むサービスサイド環境を提供するステップと、
クライアントサイドおよびサービスサイドで動作可能な動的呼出しおよびサービスインターフェイスを与えるステップとを含み、
前記クライアントサイドにおいて、
前記クライアントアプリケーションがメッセージを処理のためにランタイムに置くことができるようにするディスパッチャリクエスト関数が与えられ、かつ、前記クライアントアプリケーションがメッセージを処理後にランタイムから受けることができるようにするディスパッチャレスポンス関数が与えられ、
メッセージ処理をトランスポートから切離すクライアントリクエストトランスポート関数およびクライアントレスポンストランスポート関数が与えられ、
前記サービスサイドにおいて、
メッセージング処理をトランスポートから切離すサービスリクエストトランスポート関数およびサービスレスポンストランスポート関数が与えられ、
前記サービスがメッセージを前記ランタイムを介して受けて処理することができるようにするプロバイダリクエスト関数と、前記サービスがレスポンスメッセージを処理のためにランタイムに置くことができるようにするプロバイダレスポンス関数とが与えられる、方法。
【請求項７】
各リクエスト関数は、前記クライアントサイドおよび前記サービスサイドにおいて、対応するコールバック関数を有し、それによりメッセージ処理を非同期にする、請求項６に記載の方法。
【請求項８】
前記サービスは、ウェブサービスであり、
前記サービスプロバイダコンテナは、ウェブサービスコンテナであり、
前記クライアントアプリケーションは、ウェブサービスクライアントである、請求項６または７に記載の方法。
【請求項９】
前記クライアントサイドのアウトバウンドにおいて、
前記クライアントアプリケーションは、前記ディスパッチャリクエスト関数をコールすることによって、ターゲットサービスに対するリクエストを行ない、
前記リクエストは、アウトバウンドのクライアントサイドＳＯＡＰスタックによって処理され、
前記ＳＯＡＰスタックはクライアントリクエストトランスポート関数をコールし、
前記クライアントコンテナにおける前記クライアントリクエストトランスポート関数は、前記リクエストをトランスポートに伝え、
前記サービスサイドのインバウンドにおいて、
前記サービスリクエストトランスポート関数は前記リクエストを前記トランスポートから受け、
前記サービスリクエストトランスポート関数は前記リクエストを処理のためにインバ

ウンドのＳＯＡＰスタックに伝え、
　前記ＳＯＡＰスタックはプロバイダリクエスト関数を前記サービスによる処理のためにコールし、
　前記サービスサイドのアウトバウンドにおいて、
　前記サービスがレスポンスを与えることができる状態になると、前記サービスはプロバイダレスポンス関数をコールして前記レスポンスを処理のためにアウトバウンドのサービスサイドＳＯＡＰスタックに伝え、
　前記ＳＯＡＰスタックは前記サービスレスポンストランスポート関数をコールし、
　前記サービスレスポンストランスポート関数は、前記レスポンスを前記トランスポートに伝え、
　前記クライアントサイドのインバウンドにおいて、
　前記クライアントコンテナは前記レスポンスを前記トランスポートから受け、
　前記クライアントコンテナはクライアントレスポンストランスポート関数をコールして前記レスポンスをインバウンドのクライアントサイドＳＯＡＰスタックに伝え、
　ＳＯＡＰ処理後、前記ＳＯＡＰスタックは、前記リクエストを前記クライアントアプリケーションに送るために前記ディスパッチャレスポンス関数に伝える、請求項６から８のいずれかに記載の方法。
【請求項１０】
　リクエストヘッダに含まれるレスポンスアドレスに応じて前記レスポンスを前記クライアントアプリケーションに伝えるために、２つの異なるサービスレスポンストランスポート関数を与え選択的に呼出すことができ、
　前記クライアントサイドにおいて、
　前記クライアントアプリケーションがメッセージを処理のためにアウトバウンドのランタイムプロトコルスタックに非同期で置くことができるようにするディスパッチャリクエスト関数が与えられ、
　処理後にレスポンスをインバウンドのプロトコルスタックから非同期で受けることができるようにするディスパッチャレスポンス関数が与えられ、
　メッセージ処理をトランスポートから切離し、かつメッセージを非同期でハンドリングできるようにする、クライアントリクエストトランスポート関数およびクライアントレスポンストランスポート関数が与えられ、
　前記サービスサイドにおいて、
　メッセージング処理をトランスポートから非同期で切離し、かつプロトコルスタックがメッセージを前記ランタイムを介して受けて処理できるようにする、サービスリクエストトランスポート関数およびサービスレスポンストランスポート関数が与えられ、
　前記プロトコルスタックによるインバウンド処理後にリクエストメッセージを非同期で受けるプロバイダリクエスト関数が与えられ、
　サービスレスポンスをアウトバウンド処理のためにランタイムプロトコルスタックに非同期で置くことができるようにするプロバイダレスポンス関数が与えられる、請求項９に記載の方法。
【請求項１１】
　請求項６～１０のいずれかに記載の方法を１台以上のコンピュータに実行させるためのプログラム。